

AD-A235 585



PL-TR-91-2002

Photovoltaic Array Space Power Plus
Diagnostics Experiment

D. R. Burger

Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109

November 1990

Final Report
May 1985 - October 1990

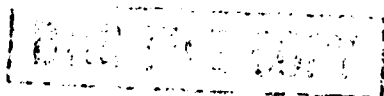
Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution	
A-1	



Approved for Public Release; Distribution Unlimited




PHILLIPS LABORATORY
AIR FORCE SYSTEMS COMMAND
HANSOM AIR FORCE BASE, MASSACHUSETTS 01731-5000




91 1 10 0 17

This technical report has been reviewed and is approved for publication.


DONALD A. GUIDICE
Contract Manager


DAVID A. HARDY
Branch Chief

FOR THE COMMANDER


WILLIAM SWIDER
Deputy Division Director

This report has been reviewed by the ESD Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS).

Qualified requestors may obtain additional copies from the Defense Technical Information Center. All others should apply to the National Technical Information Service.

If your address has changed, or if you wish to be removed from the mailing list, or if the addressee is no longer employed by your organization, please notify GL/IMA, Hanscom AFB, MA 01731. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document requires that it be returned.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
<small>Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.</small>				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE November 1990		3. REPORT TYPE AND DATES COVERED Final (May 1985–October 1990)
4. TITLE AND SUBTITLE Photovoltaic Array Space Power Plus Diagnostics Experiment			5. FUNDING NUMBERS PE 63410F PR 2822 – T01 – WU AA Contract NAS 7-918	
6. AUTHOR(S) D. R. Burger				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Jet Propulsion Laboratory California Institute of Technology 4800 Oak Grove Drive Pasadena, CA 91109			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Phillips Laboratory Hanscom AFB, MA 01731-5000 Contract Manager: Donald Guidice/PHE			10. SPONSORING / MONITORING AGENCY REPORT NUMBER PL-TR-91-2002	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION / AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) The Photovoltaic Array Space Power Plus Diagnostics (PASP Plus) experiment was designed to obtain array current-voltage [I-V] curves (indicative of power generation capability) and to provide bias voltages up to $\pm 500V$ (to test the arrays' high voltage operation). Since during the time of JPL's effort no space flight was available, instrument fabrication was limited to the brassboard stage. PASP Plus electronics consisted of the Analog Array Selection and Instrument System (ASIS) and the digital Data Acquisition and Control System (DACS). PASP Plus peripheral instruments were capable of measuring plasma electron density, sun angle, neutral particle pressure, and transient pulse characteristics of arc events. The microprocessor based controller in the DACS can have its ROMs reprogrammed to change the array selection, the number of I-V curve points, the voltage bias levels, and the sequence of data collection events; it uses the FORTH software language. The Ground Support Equipment (GSE) for PASP Plus can be used as a ground-control and data-logging substation as well as for internal calibration of ASIS and DACS and simulation of all outputs from the peripheral instruments. GSE operations are easily controlled by the use of pull-down menus. A real-time view window is provided to show operation of a serial data register and buffer.				
14. SUBJECT TERMS Photovoltaic array space power PASP plus Array high-voltage operation			15. NUMBER OF PAGES 300	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT SAR	

Prepared by the Jet Propulsion Laboratory, California Institute of Technology, for the U. S. Air Force through an agreement with the National Aeronautical and Space Administration.

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express, or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights.

Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

ACKNOWLEDGEMENTS

This work was carried out at the Jet Propulsion Laboratory (JPL), California Institute of Technology. The project was co-sponsored by the Geophysics Laboratory (GL) and the U.S. Air Force Wright Research and Development Center (WRDC) under Military Interdepartmental Purchase Request FY1455-86-N0618 through NASA Task Order RE-182, Amendment 413. Support for the development, fabrication, and testing of the experiment was provided by C. Boswell, G. Downing, M. Hanna and R. Mueller. Technical monitor support was provided by D. Guidice, GL. Acting as co-principal investigators were J. Scott-Monck (JPL) and J. Wise (WRDC). Technical advice at JPL was provided by H. Garrett, P. Leung and L. Reinhart and task management was provided by L. Sidwell, D. Burger and J. Klein.

CONTENTS

1	Introduction	1
1.1	Report Objective	1
1.2	PASP Requirements	1
1.2.1	Applicable Documents	1
1.2.2	PASP/IMPS Background	3
1.2.3	PASP Functional Requirements	8
2	PASP Plus	9
2.1	PASP Plus Functional Requirements	9
2.1.1	IMPS Documents Functional Requirements	9
2.1.2	AF PR Functional Requirements	10
2.1.3	IMPS/PASP Plus Interface Requirements	11
2.1.4	Air Force Purchase Request Attachment	11
2.2	Review of Requirements	11
2.3	PASP Plus Instrument Description	12
2.3.1	PASP Plus Electronics	12
2.3.2	GFE Solar Array Modules	24
2.3.3	Peripheral Sensors and Experiments	24
2.3.4	Flight Software	26
2.4	GSE Description	26
2.4.1	Safety and Handling	26
2.4.2	GSE Implementation	27
2.4.3	GSE Hardware Description	28
2.4.4	GSE Calibration Procedure	36
2.4.5	Flight Operation Mode	37
2.4.6	Operating and Development Software	41
2.5	Sun Sensor Acceptance Test	42
2.6	Array Acceptance Tests	42
2.6.1	Initial Array Testing	42
2.6.2	Array Retesting	43
2.7	SPICE Mission Cost and Feasibility Study	43
2.8	System Tests and Status	44
2.8.1	System Test Results	44
2.8.2	Hardware Status	44
2.8.3	Software Status	45
2.9	Pre-Ship Efforts	46
2.9.1	Pre-Ship Demonstrations	47
2.9.2	Pre-Ship Review	47
2.10	Delivery	47
3	Appendices	
A.	PASP Plus System Outline	A-1
B.	Glossary and Acronyms	B-1
C.	PASP Plus Indented Parts List	C-1
D.	PASP Plus System Test Procedure	D-1
E.	PASP Plus System Test Report	E-1
F.	PASP Plus Engineering Software	F-1
G.	PCB Power Connections	G-1
H.	Flight Software Listing	H-1
I.	Flight Software Memory Map	I-1
J.	Weight of PASP/Pegasus Experiment	J-1

K.	GSE Menu Software Listing	K-1
L.	Data Dictionary	L-1

FIGURES

2.3-1	PASP Plus Block Diagram	13
2.3.1.2.2-1	Planar Silicon Array IV Curves . . .	19
2.3.1.2.2-2	Mini-Cassegrainian Array IV Curves	20
2.3.1.2.2-3	SLATS Array IV Curves	21
2.4.5-1	HV Bias Subelement	40
3.1-1	Calculated IV Curves (DOS GRAPHICS) . .	F-3
3.1-2	Silicon Planar Array	F-4
3.1-3	Gallium Arsenide Planar Array	F-5
3.1-4	Cassegrainian Concentrator Array	F-6
3.1-5	SLATS Concentrator Array	F-7
3.2-1	Mini-Cassegrainian Tilt Axis	F-10
3.2-2	Mini-Cassegrainian Rotation Axis	F-11
3.3-1	Orbit Pen Plot	F-13

PHOTOGRAPHS

1	Langmuir Probe Sweep	E-35
2	HVPS, Set #1, 50V Noise	E-35
3	HVPS, Set #1, 300V Noise	E-35
4	HVPS, Set #1, 500V Noise	E-35
5	HVPS, Set #2, 50V Noise	E-36
6	HVPS, Set #2, 150V Noise	E-36
7	HVPS, Set #2, 300V Noise	E-36
8	HVPS, Set #2, 500V Noise	E-36
9	In-Rush Current	E-37
10	Input Noise	E-37
11	Input Noise	E-37
12	Input Noise	E-37

TABLES

2.4.5-1	Module Test Matrix	38
2.4.5-2	Data Acquisition Element	38
3.5-1	TPM Packet Structure	A-3
4.1-1	PASP Plus System Commands	A-4
4.1.2.2-1	IV Curve Load Selection Commands	A-6
4.1.2.3-1	Bias Voltage Selection Commands	A-7
5.9-1	Telemetry Packet Format	A-11
5.9.1.2-1	MET Time Code	A-12
2.8.10.1-1	Housekeeping Voltages Ranges	D-6
2.8.10.1-1	Housekeeping Voltage Ranges	E-14
4.3.1.1-1	Array Mounting Torques	E-25
3.4-1	Table of Gray Code Bytes and Angles . .	F-31

THE PASP Plus EXPERIMENT

1 Introduction

1.1 Report Objective

The objective of this final report is to summarize the five years of hardware development and fabrication represented by the Photovoltaic Array Space Power Plus Diagnostics (PASP Plus) Instrument. This report presents the original PASP Experiment requirements and background and then the modifications which were requested to transform the PASP Experiment into the PASP Plus Instrument. The PASP Plus hardware and software is described. Test results for components and subsystems are given as well as final system tests. Also included are appendices which describe the major subsystems and present supporting documentation such as block diagrams, schematics, circuit board artwork, drawings, test procedures and test reports.

1.2 PASP Requirements

1.2.1 Applicable Documents

The following list of applicable documents is culled from the IMPS Project list of documents and was considered to be the guiding set of documents for the original PASP Experiment.

1.2.1.1 Jet Propulsion Laboratory

CS50276	Detail Requirements for Fasteners, External Threaded, Nonmagnetic Heat and Corrosion Resistant Steel
DM509306	Design Requirement, Electronic Equipment and Cabling Design and Fabrication Requirements and Processing Techniques
D-2836	Policies and Requirements for IMPS Investigations (PAR), Vols. I & II, dated December 1987
D-2837	Investigation Requirements Document (IRD) for the IMPS-1 Payload, dated March 1986
D-2838	SPAS/IMPS-1 Interface Control Document (ICD), Ver. 2.1, dated 21 January 1987
601-4B	Flight Project Safety Guidelines and Requirements
PD 625-231	High Margin Fracture Mechanics Screening

Criteria

STD00009	Engineering Standard, Preferred Materials, Fasteners, Processes, Packaging, and Cabling Hardware
ZPP-2061-PPL	Preferred Parts List
(unreleased)	Interactions Measurement Payload for Shuttle (IMPS-1) Mission Plan (Preliminary), dated September 1986

1.2.1.2 National Aeronautics and Space Administration (NASA)

MIL-STD-975F	NASA Standard Electrical, Electronic and Electromechanical (EEE) Parts List
MSFC M&P-STD-EH13- INDE	Guidelines and Requirements for Fracture Control Programs
MSFC-SPEC-522A	Design Criteria for Controlling Stress Corrosion Cracking
NASA NHB 8060.1B	Flammability, Odor, and Off-Gassing Requirements and Test Procedures for Materials in Environments which Support Combustion
NHB 5300.4 (3A-1)	Requirements for Soldered Electrical Connections
NHB 5300.4 (3H)	Requirements for Crimping and Wire Wrap
NHB 5300.4 (3I)	Requirements for Printed Wiring Boards
PPL-17	GSFC Preferred Parts List

1.2.1.3 Military

MIL-M-38510	Microcircuits, General Specification for
MIL-P-13949	Plastic Sheet, Laminated, Metal Clad (for Printed Wiring Boards)
MIL-P-18177	Plastic Sheet, Laminated, Thermosetting, Glass Fiber Base, Epoxy-Resin
MIL-P-81728	Plating, Tin-Lead, Electrodeposited and

Fused

MIL-P-556736

For Prepregnated Materials

MIL-S-19500

Semiconductor Devices, General Specification for

MIL-STD-1522A

Design Criteria for Controlling Stress Corrosion Cracking

1.2.1.4 Federal

QQ-S-571E

Federal Specification, Solder, Tin Alloy, Tin-Lead Alloy, And Lead Alloy

1.2.1.5 Rockwell International

SOD-79-0280

Advanced Crack Propagation Predictive Analysis Computer Program "FLAGR04"

1.2.1.6 U.S. Air Force (DOD)

MIL-STD-1574A

System Safety Program for Space and Missile Systems

1.2.2 PASP/IMPS Background

This section provides the background underlying the objectives and requirements of the PASP experiment as formulated during the IMPS Project period. This background is needed to lay the foundation for the following functional requirement section and the inauguration of the PASP Plus experiment.

1.2.2.1 IMPS Project Background

The Interaction Measurements Payload for Shuttle (IMPS) mission, was a JPL project for the United States Air Force. The IMPS Project was started with the intent of providing a low cost, reflyable measurement platform for engineering and science experiments in the near-Earth, polar environment. Shuttle launches were to be made from the Vandenberg (Western Test Range) facility in order to achieve high inclination. A variety of instruments and experiments were to be flown on the first and subsequent IMPS missions. The first IMPS mission (IMPS-1) was originally scheduled for launch in December 1987 and had six major experiments: Space Based Radar, Optical Effects Module, Surface Potential Monitor, Environmental Interaction Monitors, Transient Pulse Monitor and Photovoltaic Array Space Power (PASP). All of these experiments were to be fitted on board a Shuttle Pallet Satellite (SPAS) made

by Messerschmitt-Bolkow-Blohm. All of the experiments were designated as Class C under the JPL Defense and Civil Programs Standard for Flight System Quality Classifications (JPL D-2547).

Launch of the IMPS-1 mission was placed on hold after the January 1986 Challenger accident. It was later decided that the IMPS mission could no longer be supported although the PASP experiment would be continued due to its value. In March of 1988 the IMPS Project was officially closed at JPL. During the period of uncertainty from January 1986 to March 1988 the PASP experiment was continued on a reduced level since there was no defined experiment carrier.

1.2.2.2 IMPS Mission Design Philosophy

The IMPS mission, as conceived by the Air Force Geophysics Laboratory (AFGL), was a platform for performing measurements in the near-Earth environment at high inclinations. The design goal was to develop a multimission capability to perform measurements of the aurora/polar regions for the Department of Defense (DOD) which can effectively be used to study the impact of this environment on future polar missions. The engineering and science data returned by IMPS would provide a benchmark for the evaluation of new spaceborne engineering systems including planned extravehicular activities (EVA) by astronauts.

Data acquired from the IMPS mission would have come from five separate, preintegrated packages of instruments. In the design of the IMPS mission it was important that consideration be given to the needs of individual instruments and the synergisms that result from the combined measurements of an identical environment by a number of different instruments. Therefore consideration must be given in the design to a modular development strategy whose fundamental units have a separate integrity that, in contributing to the overall design, have the additional characteristic of being units that can be combined in new ways to achieve new and unforeseen goals.

1.2.2.3 Experimenters Working Group Meetings

PASP design activities had to be modified after January 1986. Before the Challenger accident the IMPS Project had a revised launch date of mid-1988, while after the accident, the launch date was uncertain. Prior to January 1986, the number of experimenters was such that there was a need for coordination meetings. Preliminary design reviews for the various experiments were also held during this period.

There were a total of five Experimenters Working Group (EWG) Meetings. The minutes for these meetings are not included in this final report since much of the information is not pertinent to PASP and even less is pertinent to PASP Plus. Summarized below are the

PASP relevant portions of the minutes of the EWGs.

1.2.2.3.1 First Experimenters Working Group Meeting

The First EWG Meeting was held on 22-23 January 1985 at AFGL. There was some discussion of instrument and sensor location on the SPAS carrier. PASP was given an action item to further define data transmission rate requirements.

1.2.2.3.2 Second Experimenters Working Group Meeting

The Second EWG Meeting was held on 25-26 April 1985 at JPL. The PASP sun sensor was mentioned as a source of sun angle information for spacecraft attitude correction. PASP timeline has been roughed out and will be in the free-flyer mode after SPAS has stabilized. There was some discussion of data analysis but no requirements were laid upon PASP.

1.2.2.3.3 Third Experimenters Working Group Meeting

The Third EWG Meeting was held on 31 July-1 August 1985 at AFGL. Location of the PASP electronics box in relation to the Space-Based Radar experiment has become an issue. Action items were created for PASP generated fields, materials, and characteristics of the emitter. A power budget of 500 watt-hours was requested for PASP. A PDR for PASP was scheduled for October 1985 and a CDR for March 1986. Delivery of PASP was scheduled for March 1987.

1.2.2.3.4 Fourth Experimenters Working Group Meeting

The Fourth EWG Meeting was held on 28-29 January 1986 at JPL. The IMPS Project was rescheduled for delivery of instruments (including PASP) by March 1988. There were concerns about both weight and power. Explosion of the Challenger was announced part way through the meeting. The PASP PDR was to be held shortly after the EWG meeting. PASP array mounting has been split up into two mounting plates. A request was made for an additional 154K of funding for PASP. Leakage current for large arrays was calculated to be as high as 100mA. The spacecraft potential was calculated to be as much as -400 V unless an emitter is used. The PASP electronics were now designed to be housed in two boxes and placed inside the SPAS frame.

1.2.2.3.5 Fifth Experimenters Working Group Meeting

The Fifth EWG Meeting was held on 11-12 June 1986 at AFGL. It was announced that the first practical launch opportunity would be in late FY 1990. A major item for all experimenters was parts approval. Development of the PASP experiment is presently ahead of the development of all of the other experiments. Another EWG was planned for late October 1986 (but was never held). An action item was created for IMPS to provide parts guidelines to experimenters.

AFWRDC was asked to declassify the PASP arrays. There were problems with a lack of interface data and other IMPS project information needs mainly due to uncertainty about the procurement of a SPAS carrier.

1.2.2.4 PASP Experiment Objectives

The objective of the PASP experiment was to design, develop and fabricate a flight instrument package that, when rebuilt, repackaged, qualified and operated in conjunction with a number of conventional and potentially survivable Air Force array designs, would be able to provide engineering and scientific information concerning the influence of the low- to mid-altitude space environment on solar array performance.

There were three main data objectives for the PASP experiment. The first, was to determine the negative bias voltage where the test array modules discharge or arc. The experiment was to determine this as a function of the space plasma density. The transient pulses from arcs would also be measured.

The second objective was to measure the array modules' current leakage to the space plasma, a potential power loss factor, as a function of positive bias voltage. The final objective was to measure the power output (current-voltage characteristics) of each array under actual (illuminated) operating conditions.

The PASP experiment objective then was to investigate the effect of the plasma environment on array modules that are producing power under full solar illumination. A unique characteristic of this experiment involves the solar array test modules. The two concentrator array modules are predominantly composed of exposed conducting surfaces, which are or are not biased during the experiment. The remaining modules are representative, with respect to the amount of exposed conducting surface, of state-of-the-art planar solar arrays now considered for Air Force missions.

1.2.2.5 PASP Experiment Background

The purpose of the PASP experiment was to measure the effect of the low- to mid-altitude space environment on the performance of conventional and potentially survivable solar array configurations. The PASP experiment was originally scheduled for completion by March 1987. This plan had to be modified due to problems in obtaining a suitable flight for the experiment. Since a known carrier interface was not available, the deliverable instrument was changed from fully qualified flight electronics enclosures to unencased, non-flight brassboards and associated cabling and ground support equipment. Similarly, the photovoltaic arrays would not be mounted on array plates.

The instrument was designed to measure the surrounding space plasma

and partial pressure environment, orient and then measure the current-voltage curves of the various solar array modules, deliberately bias each array to high voltages (+/- 500 volts), measure the plasma leakage current and the array generated hard wire and radiated electromagnetic interference as well as properly sequence and control the various tests and transfer the data to the carrier for recording and downlinking.

The arrays were to consist of four to six designs: two or three conventional and two or three potentially survivable. They were to represent various elements of the space-power technology program including GaAs solar cells and solar-array concentrator designs.

A test sequence was planned that, if permitted by the host-spacecraft orbit, additional data on the effects of the auroral substorm environment could be obtained, as well as information that could be used to support the results derived from previously flown similar experiments such as: SCATHA, PIX-I (Plasma Interaction Experiment flown on 5 March 1978) and PIX-II experiments. Other plasma interaction experiments had been planned: SPHINX (space plasma-high voltage interaction experiment) was launched in February 1974 but did not collect data due to booster failure; VOLT-A (Voltage Operating Limit Tests), a NASA Lewis experiment, was a planned mission (now shelved) which would have covered some of the technology issues that are covered by PASP.

1.2.2.6 Preliminary Design Review

A Preliminary Design Review (PDR) for the PASP Experiment was held at JPL on 30-31 January 1986 immediately following the Fourth EWG meeting. The PASP PDR was the first PDR for any IMPS experiment and as such was heavily attended and scrutinized by JPL management. Results of the PDR were favorable although there was some concern about the future due to the 28 January 1986 explosion of the Challenger.

1.2.2.7 Critical Design Review

The Critical Design Review (CDR) was held at JPL on 24-25 June 1987 approximately 18 months after the PDR. Issues raised by the CDR Board were: resource allocation (power, mass and volume), high voltage inhibits for shuttle safety, structural analysis and functional verification testing. A total of 10 action items were levied by the Board with six of the items levied against PASP. The six PASP action items were: 1) low voltage power supplies were undefined, 2) a grounding tree was needed, 3) a rationale was needed for no array thermal-vacuum testing, 4) the stress analysis and fracture control reports were needed, 5) a better definition was needed of the high voltage hazard, 6) and special inhibits were needed on high voltage operation.

At the time of the CDR the mission design effort by IMPS had

already been terminated which raised some integration questions. All of the action items were subsequently addressed except that the grounding tree was never completed and the high voltage inhibits were never implemented due to lack of carrier. It should be noted that there was both a calibration test plan and a functional test plan prepared before the CDR.

1.2.2.8 Safety Review

An IMPS mandated safety review was conducted by the PASP task manager on 28 August 1985. Following this review a Safety Package covering Phases 0 and 1 was prepared and submitted to the IMPS Project Office (Reference: IOM #3464-86-252, dated 13 November 1986, from L. Sidwell to H. Eyerly). This safety package was shuttle launch oriented and raised the following issues:

- a. EMI noise from arcing may interfere with control
- b. Sharp edges are present in concentrator array optics
- c. High voltage is present during ground operation

1.2.2.9 FMEA Review

A Failure Modes Evaluation and Analysis (FMEA) Review was required on the PASP/IMPS avionics interface (Reference: JPL IOM #HWE:845-87-024, from H. Eyerly to Gary Hill, et. al. dated February 24, 1987). This review was to show that any one component or circuitry failure which occurred on the PASP side of the interface would not propagate to the IMPS side of the interface. This review process was not completed due to lack of interface definition (Reference: JPL IOM #HWE:845-87-069, from H. Eyerly to Pete Theisinger, dated July 7, 1987).

1.2.3 PASP Functional Requirements

The functional requirements laid upon the PASP experiment were often driven by the needs of the IMPS project rather than the need to perform the PASP experiment. Changes to these functional requirements are detailed below in section 2.1.

2 PASP Plus

In April 1988 the PASP Experiment was redirected to a non-flight status and additional instrument support requirements were added. Funding was reduced to reflect the non-flight status and the delivery date was accelerated to September 1989. The experiment was officially renamed the Photovoltaic Array Space Power Experiment Plus Diagnostics (PASP Plus).

2.1 PASP Plus Functional Requirements

The change from a shuttle flight mission for PASP to an undefined mission brassboard design for PASP Plus caused a number of modifications to the functional requirements contained in the four PASP requirements documents. The following document sections have been modified as shown from the PASP requirements to the PASP Plus requirements. If a document section is not shown below then there was no modification of that particular requirement. The listed documents themselves were never modified to include any functional requirements changes since the issuing authority, the IMPS Project Office, was no longer in existence.

2.1.1 IMPS Documents Functional Requirements

IMPS-1 Mission Plan

- 2.2.3 This requirement has been dropped for safety reasons.
- 3.2.2 The effects of arc discharges, contamination, and oxygen erosion on exposed surfaces will be studied only if the carrier is recoverable.
- 3.2.3.1 Transient Pulse Monitor (TPM) The TPM will record electrical transients on the PASP Plus system and cables.
- 3.2.11 Pressure Gauge (PG) The PG will measure the spatial and temporal variations of the neutral gas density within the region of the PASP Plus system.
- 3.3.2 PASP Plus orbit may be different from the IMPS-1 orbit. The PASP Plus/spacecraft bus relative orientation shall be maintained such that the spacecraft bus does not cast a shadow on the PASP Plus instrument and that reflection of sunlight from the spacecraft bus onto the PASP Plus arrays is minimized. c) The PASP Plus requires that the spacecraft bus be oriented out of the FOV of the test arrays. d) Same as c) above. e) The PASP Plus arrays must be aligned toward the Sun within 10 sec after sunrise in order for valid data to be acquired on the concentrator arrays.

3.5.2.1 Mode characteristics are TBD.

IMPS PAR, Vol. I

7.10 Functional Test Plan (see text below).
7.11 Calibration Plan (see text below).
9 Deliverables

IMPS PAR, Vol. II

The following standards manual was used as the applicable document: MIL-STD-975G, 1 October 1986.

IMPS-1 IRD

Table 2-2 Behavior of biased solar cells in Polar environment: electrometer range is from 10^{-6} to 10^{-3} A.

2.1.2 Investigation description (see text below).
3.2 PASP requirements (see text below).
Appen. C PASP telemetry data format (see text below).

SPAS/IMPS ICD

5 Electrical Power Interface (see Figure below).
7 Isolation, Grounding and Bonding (see Figure below).

2.1.2 Air Force Purchase Request Functional Requirements

1. The objective of the Photovoltaic Array Space Power Plus Diagnostics (PASP Plus) Experiment is to develop, design and fabricate a brassboard version of an instrument package that will operate in conjunction with different advanced-concept Air Force solar array modules to provide scientific and engineering information concerning interactions between the low-earth polar orbit plasma environment and high-voltage solar arrays.
- 2.2 References to the STS as a carrier are no longer appropriate - this includes all of the noted references.
- 3.2.4 SPAS interface is no longer valid.
- 3.2.6 EIP level of testing is no longer valid.
- 3.2.9 Protoflight testing is no longer valid.
- 3.3 AFGL/PHE Responsibilities
Note: This section was added to incorporate additional information.
- 3.3.1 Provide a Cold Cathode Pressure Gauge as GFE.
- 3.3.2 Provide a Transient Pulse Monitor and Sensors as GFE.
- 3.3.3 Attend meetings and accomplish action items as

- appropriate.
- 4.1.2.1 The Integral Cover Array has been dropped from the program.
 - 4.2.3.1 Actual size of the SP array is 50cm x 25cm.
 - 4.2.6.1 Modules will be shorted while IV data is not being taken in order to reduce plasma charging effects.
 - 4.2.6.2 Temperature measurement devices are platinum RTDs for improved reliability.
 - 4.2.6.4 Plasma-coupling current will only be measured over the range of 10^{-6} to 10^{-1} A. Reduction in the low current range is required due to circuit component noise. Reduction in the high end current is due to power handling difficulties at 500 V.
 - 4.2.6.5 Arcing events will be counted by the TPM instrument not by PASP.
 - 4.2.6.9 The requirement for -300 V bias during occultation was changed to -200 V for 20 seconds every forty seconds (20 seconds on and 20 seconds off).
 - 4.2.7 Same as above.
 - 4.2.9 Mounting on SPAS is no longer valid.
 - 4.2.10 Same as above.
 - 4.2.11 Pre-flight testing is no longer valid.
 - 5.4 Delivery of complete mechanical package is no longer valid.

2.1.3 IMPS/PASP Plus Interface Requirements

Since there are no specific carrier interface requirements for PASP Plus it was decided to continue using the PASP requirements.

2.1.4 Air Force Purchase Request Attachment

- 7.1.1 While the PASP Plus experiment will operate with a sun pointing accuracy of $\pm 1.5^\circ$ it is better to try to keep this accuracy to $\pm 1^\circ$.
- 7.1.2.2 There are three data acquisition elements but the detailed data rates and timing have been modified as required. The requirement to hold the SP array to 0 V is contradictory to the earlier purchase request requirement of -300 V and the subsequently agreed upon -200 V.

2.2 Review of Requirements

While the PASP Plus program did not go through as many modifications as some programs, it is best, at this point, to review the requirements which must be met by the delivered equipment.

The basic requirements have already been stated in the PASP Functional Requirements section above as modified by the PASP Plus

Functional Requirements section also above. These two sections reflect those requirements shown in various IMPS and Air Force documents as well as some interface requirements generated in the course of development.

2.2.1 Requirements which have been added or changed are:

2.2.1.1 A thermionic emitter has been added.

2.2.1.2 Housekeeping circuits have been added to monitor power supply voltages, heat sink temperatures, and circuit board temperatures.

2.2.1.3 The PASP Plus experiment electronics are not capable of reaching bias voltages of more than 500 V.

2.2.2 In summary it can be said the the completed PASP Plus experiment meets the original functional requirements as modified by various agreements with the Air Force and the necessities of a modified mission.

2.3 PASP Plus Instrument Description

The PASP Plus Diagnostics Instrument consists of the following components: PASP Plus electronics, GFE solar arrays, peripheral sensors and experiments, GSE, and associated flight and GSE software. Relationship of the components is illustrated by a block diagram, Figure 2.3-1. The peripheral sensors are a sun sensor procured by JPL and a GFE Langmuir Probe. The peripheral experiments along with their sensors are the GFE Transient Pulse Monitor and Cold Cathode Pressure Gauge which were added to the original PASP Experiment to make PASP Plus. The PASP Plus GSE is described separately in the next section. The flight software is described at the end of this section and the GSE software is described at the end of section 2.4.

2.3.1 PASP Plus Electronics

The PASP Plus electronics consists of two subsystems (DACS and ASIS) and some housekeeping sensors. Each of the subsystems has its own power supply board and housekeeping board along with other circuit boards. Because of packaging considerations the power supply and housekeeping boards for the DACS subsystem are in the same cold plate/card cage as the ASIS circuit boards. The PASP electronics were designed for use at atmospheric pressure to avoid internal high voltage arcing. Each of the sub-systems has an attached pressure sensor (PS1 for ASIS and PS2 for DACS) to monitor container pressure. Additional housekeeping data is obtained from temperature sensors mounted on the housekeeping circuit boards (TM1 for ASIS and TM3 for DACS) and the power supply circuit board heat sinks (TM2 for ASIS and TM4 for DACS).

2.3.1.1 Data Acquisition and Control System (DACS)

The DACS for PASP Plus is a system based on twelve plug-in printed

JPL PASP PLUS BLOCK DIAGRAM

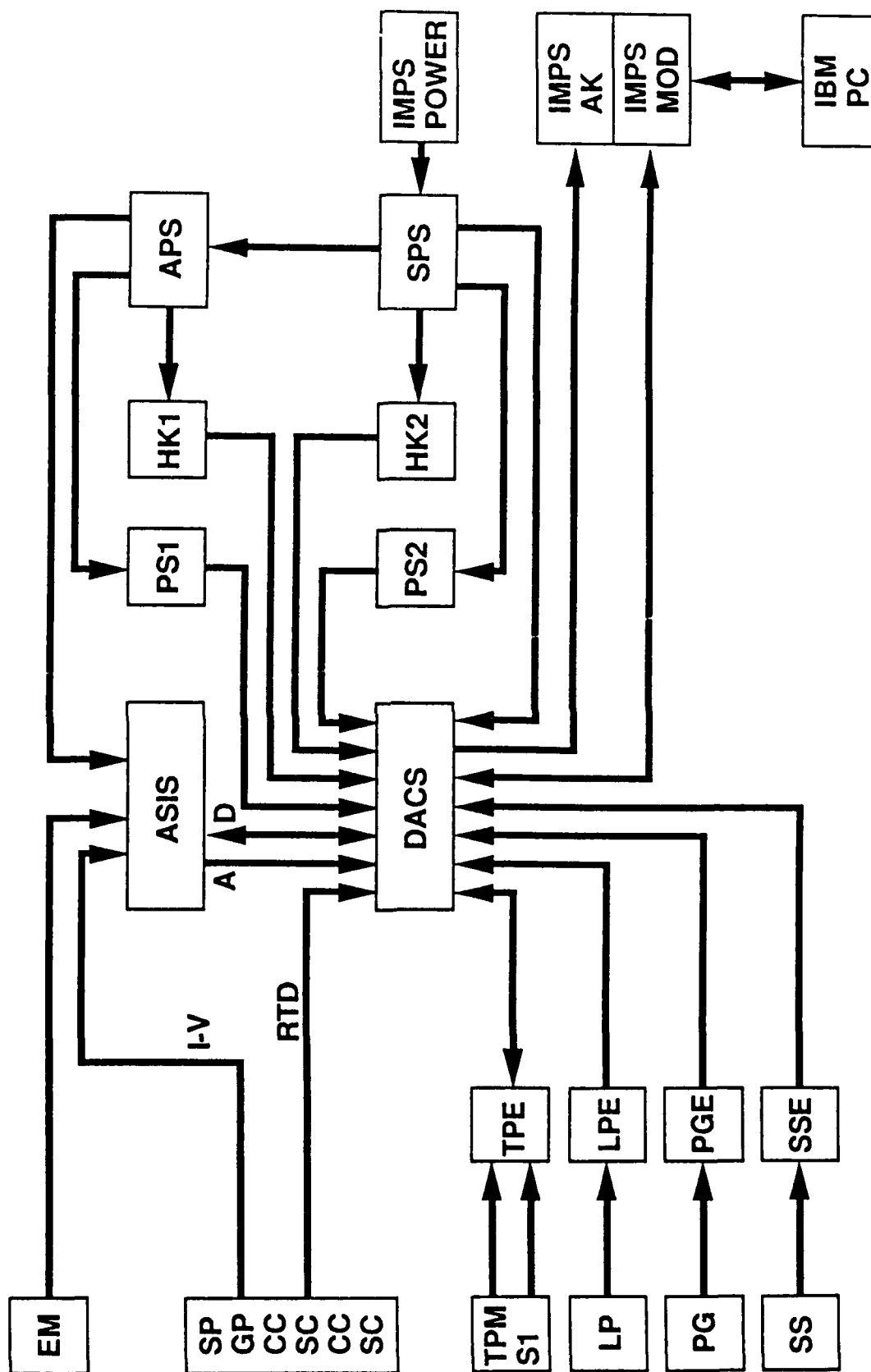


Figure 2.3-1

circuit boards packaged with a mother-board in a ruggedized cold plate/card cage and two additional printed circuit boards (house-keeping and power supply) co-located with the ASIS electronics. The cold plate/card cage approach has been used before by JPL and other aerospace companies for shuttle flights. The printed circuit boards and card cage are designed to resist damage by the Shuttle launch and flight environment and provide a path for heat transfer between the electronics and the carrier. The twelve DACS printed circuit boards are discussed below in the same sequence that they are placed in the card cage (reading from left to right with the mother board to the rear).

2.3.1.1.1 The microprocessor card (SEQ) contains a Harris 80C86 MPU. The SEQ card also contains 2K bytes of system and buffer RAM and 8K bytes of program PROM. The software development effort was reduced by the use of these built-in hardware features. The SEQ card contains the logic functions for bus control, reset control, interrupt control, bus driver, and system clock generation.

2.3.1.1.2 The digital I/O card (DIG) is used to input digital data and commands to the PASP Plus subsystems. The commandable instrumentation subsystems consist of array selection and IV measurement, the high voltage power supply, the Langmuir probe, the emitter, and the TPM instrument.

2.3.1.1.3 The temperature sensor card (RTD) reads the temperatures of the test solar array assemblies. The platinum RTD sensors are pulsed and read on command from the DACS.

2.3.1.1.4 Two analog-to-digital (AD1 & AD2) cards are required. Each card contains two 8-channel MUX chips which feed two A/D convertors to provide sixteen different ± 5 V input channels with up to 12-bit outputs. Each card collects analog data from instrument subsystems and also housekeeping and status data.

2.3.1.1.5 The low power mode control and serial command card (CMD) contains the primary command decoder and low voltage power supply controller to receive commands and control the status of the instrument. Also on this card are the watchdog timer, latch-up sense and recovery circuitry, and carrier interface isolation circuits.

2.3.1.1.6 The sun sensor card (SSI) contains two separate sun sensor interface control and isolation circuits which provide sun sensor data to the carrier and to PASP.

2.3.1.1.7 Two synchronous serial I/O (RDO & TDO) cards are used to communicate with a carrier through optically isolated interfaces. Serial output data is sent to the carrier. One card (RDO) provides real time data to the telemetry down-link while the other card (TDO) provides non-real time data to the carrier tape recorder.

2.3.1.1.8 The mission elapsed time (MET) card contains the clock and control circuits and corresponding carrier interface isolation circuits to process any carrier inputs and provide the necessary synchronization with the PASP Plus DACS clock system.

2.3.1.1.9 The TPM interface card (TPI) contains the electronics necessary to convert and channel incoming commands for proper TPM operation.

2.3.1.1.10 The terminator card (TER) provides proper impedance termination for the mother-board signal bus lines.

2.3.1.2 Array Switching and Instrumentation System (ASIS)

The ASIS subsystem circuit boards do not plug into a mother-board because many of them carry high voltages or high currents. The ASIS subsystem consists of circuitry which accepts and decodes digital signals from the DACS in order to generate requested high bias voltages, and switch arrays, IV curve loads and emitter. Also located in the ASIS card cage package are two power supply boards (APS & SPS) and two housekeeping boards (HK1 & HK2). The APS and HK1 circuit boards are for the ASIS sub-system and the SPS and HK2 boards are for the DACS sub-system.

2.3.1.2.1 Relay Switching and Logic Unit - The ASIS relay switching and logic unit consists of two circuit boards - the logic board and the relay board. The logic board (RLI) controls operation of the emitter relay and the other circuit boards in the ASIS based upon commands from the DACS. Digital logic is used to provide isolation, fault protection and to drive relays.

The relay board (RLR) is used to switch arrays to the IV curve measurement boards, allow application of high voltage bias, and switch concentrator frames from being isolated to being connected to the array negative lead.

2.3.1.2.1.1 RLI Board - Three different kind of digital commands from the DACS are received by the logic board: mode, IV board select and IV curve load select. The logic board also generates a signal which activates the high voltage power supply (HVPS +15V CONT OUT) by turning on the +15V AD210 power to those boards.

Mode Commands - There are four different mode commands received by the opto-isolators, U3 and U4: -IV ON, -HV ON, -EMIT, and -POL. These commands are used to generate five different modes: IV, HV, HVT2, EMITTER and REV POL.

The -IV ON command means that the input to opto-isolator U4 goes low which turns off the output. This signal is inverted into a high signal by the hex inverter, U7, which feeds the lowest select input of the 3-to-8 decoder, U9. If the other two select inputs of the decoder are low (the third select is always low since it is grounded) then the Q1 output goes low, meaning the IV mode is on.

The IV mode signal performs two functions. It sets both enables of the octal buffer/line driver, U5 to low, and it sets one input low for six NOR gates in U12 and U13. Obtaining an output from these devices is thereby dependent on the presence on the -IV ON command.

The -HV ON command also turns off an opto-isolator output which performs three functions. It sets one input of the emitter NOR gate in U14 to low; it is inverted into a high signal by U7 and then it sets one input of the AND gate in U6 and feeds the second select input of the decoder, U9. If the other two selects are low then the Q2 output goes low, meaning the HV mode is on. The HV mode signal sets one input low for six NOR gates in U11 and U12. Obtaining an output from these devices is thereby dependent on the presence of the -HV ON command.

A combination of the -IV ON and -HV ON commands is used to create the HVT2 mode. This mode is used to apply the high voltage bias to the concentrator arrays when their frames are connected to the negative side of the array. The additional logic to achieve this switching is included in the NOR gates of U14 and the OR gates of U15.

The -EMIT command only sets to low the other input of a U14 NOR gate. If the -HV command has also been given then both inputs of this gate are low and the EMITTER ON signal is generated. This signal is sent to the Darlington driver, U8, where it completes the return line for the +24V RLY1 supply through the coil of the relay, K1. When K1 is activated it sends power to the emitter filament. Any emitter emission current flow is sensed by R3 and sent to an A/D convertor in the DACS through the isolation amplifier, U19, as the EMITTER OUT signal.

The -POL command is simply inverted to a high signal by U7 and split into two REV POL OUT signals going to J2 and J3. The J2 output (K5) goes to the RLR board where it inverts the leads on the two planar arrays to provide a series aiding circuit. The J3 output goes to the HVE board where it inverts the output from the high voltage oscillator board.

IV Board Select Commands - The IV board select commands are contained in three bits received by the opto-isolator, U4. Commands which are all zeros or all ones are not used. The other commands make up a set from 1 to 6 and are used to turn on the respectively numbered IV boards. The three command bits are inverted to high signals by U7 and sent to the three select inputs of the decoder, U10. The Q1 through Q6 low signal outputs from U10 are then used to set the other input of the NOR gates in U11, U12 and U13. If the IV mode has already been commanded then the U12 and U13 NOR gates generate the required IV board output signal which is sent to connector J3 and to the XOR gates of U16, U17 and U18. For example, when array 1 is selected in the IV mode (the IV1 circuit board), the A gate of U18 produces a signal, K1. When

array 1 is selected in the HV mode two signals are produced, K1 and K2. If both the IV and HV mode are on then the K1 signal is not produced.

IV Curve Load Select Commands - The IV load select commands are contained in eight bits received by the opto-isolators, U1 and U2. These bits are simply sent to a buffer, U5, where they are then sent out as signals to connector J3 if the IV mode command has also been received.

2.3.1.2.1.2 RLR Board - There are eighteen double-pole, double-throw relays on the relay board. Some relays are dedicated to just one array and some use one pole for one array and one pole for another. The switching requirements for planar arrays are different from the requirements for concentrator arrays and are detailed below.

Planar Array Switching - Relays K1-K5 are used to switch the SP and GP arrays (arrays 1 and 2). Relay K1 must be activated to switch the SP array from a shorted and grounded condition to the next relay in the circuit, K2. When the K2 relay is inactive it connects the array positive and negative leads to the IV1 circuit board for IV curve generation. When the K2 relay is activated it switches the array leads to the K5 relay which is connected to the high voltage power supply. The logic of the K2 relay thus prevents the array from being simultaneously connected to the IV measurement circuitry and the high voltage supply. The K5 relay is used to switch the application of high voltage from the negative lead of the SP array (K5 inactive) to the positive lead of the array (K5 active). This last switching function is activated by the REV POL OUT signal from the RLL board and is needed to allow the high voltage bias to be applied in a series aiding fashion. When the high voltage bias is positive it is added to the negative lead so that the array negative terminal may be biased as high as 500 V. Since the SP array generates about 100 V the positive terminal of the array can then be as much as 600 V above the high voltage supply ground. The GP array is treated in the same way as the SP array except relays K3 and K4 are used for array switching. The other pole of the K5 relay is used for the GP high voltage switching.

The other planar array (array 5) was originally planned to be a small array which generated a small voltage. This array was not wired to provide series aiding switching. Instead, only two relays, K10 and K11, are used. Activating K10 switches array 5 to the IV5 circuit board. Activating K11 allows the high voltage bias to be applied to the negative lead of the array. All three of the concentrator arrays are also low voltage arrays and are handled in much the same way in so far as high voltage bias is concerned.

Concentrator Array Switching - The concentrator arrays require more switching than the planar arrays because they have three leads -

the positive and negative array leads and the array frame lead. Switching of the CC array (array 3) uses relays K12, K14, K15, and K16. When K14 is activated it switches the CC array to the IV3 circuit board. When K16 is activated it allows the high voltage bias to be applied to the positive lead of the array. When K15 is activated the CC array frame is switched from being grounded to having a 1 megohm resistor to ground. The K15 relay is activated the same time as the K14 relay in order to bring the array from a grounded state to a measurement state. When the K12 relay is activated the CC array frame is tied to the negative lead of the array to prevent plasma from biasing the frame by charging. Tying the frame to the negative lead is the reason for applying the high voltage bias to the positive lead of the concentrator arrays. In a similar manner the other two concentrator arrays (SC and array 6) are switched using the following two relay groups: K12, K13, K17 and K18; and K6, K7, K8, and K9 respectively.

2.3.1.2.2 Array IV Curve Measurements - IV curves and temperature measurements can be acquired on each of six solar array modules during flight. The temperature measurements are performed using a platinum RTD mounted to the back of each module and read by the DACS subsystem. The IV curves are acquired on each of the modules by means of six independent IV circuit boards (IV1-6) for maximum reliability. These circuits consist of a parallel ladder of eight relay switched resistors capable of providing up to 256 individual IV data points. These IV circuits are read by and are under the programmed control of the DACS with the signals being buffered by the RLL board. Module voltage is read directly across the resistor network while module output current is measured as a voltage across a sensing resistor in series with the load network. Thus, each resistor network is driven by the DACS to present a pre-programmed number (32 points are presently used) of variable impedances to each module. This IV data and the temperature data can be used to calculate IV curves from open circuit voltage through the maximum power point to short circuit current over the entire temperature range experienced by the modules during orbit. Figures 2.3.1.2.2-1 through 2.3.1.2.2-3 show IV curves obtained during system simulation and testing. These curves are described below.

The highest curve (square data points) represents a composite full sunlight curve generated by the IV simulator using engineering software calculated data points (see Appendix F) for the highest expected voltage (at -100°C) and the highest expected current (at +100°C). The lowest curve (rhombus data points) represents the simulator generated occultation (dark) condition. The laboratory functional test curve (triangle data points) was created using floodlamps and is also rather low. The last curve (x data points) was made in full sunlight after the functional tests were completed. The values given on the IV curve figures are simply raw data values from 0-255 as taken from the System Test Report included as Appendix E.

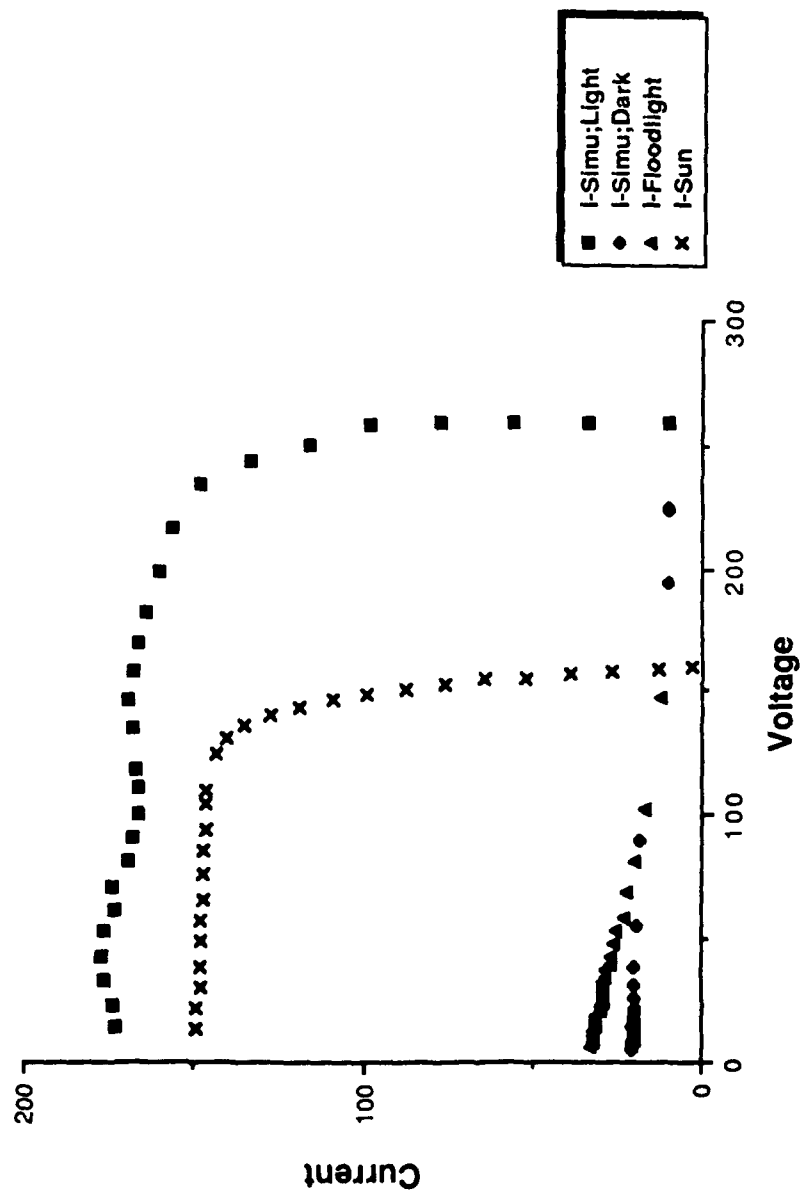


Figure 2.3.1.2.2-1 Planar Silicon Array IV Curves

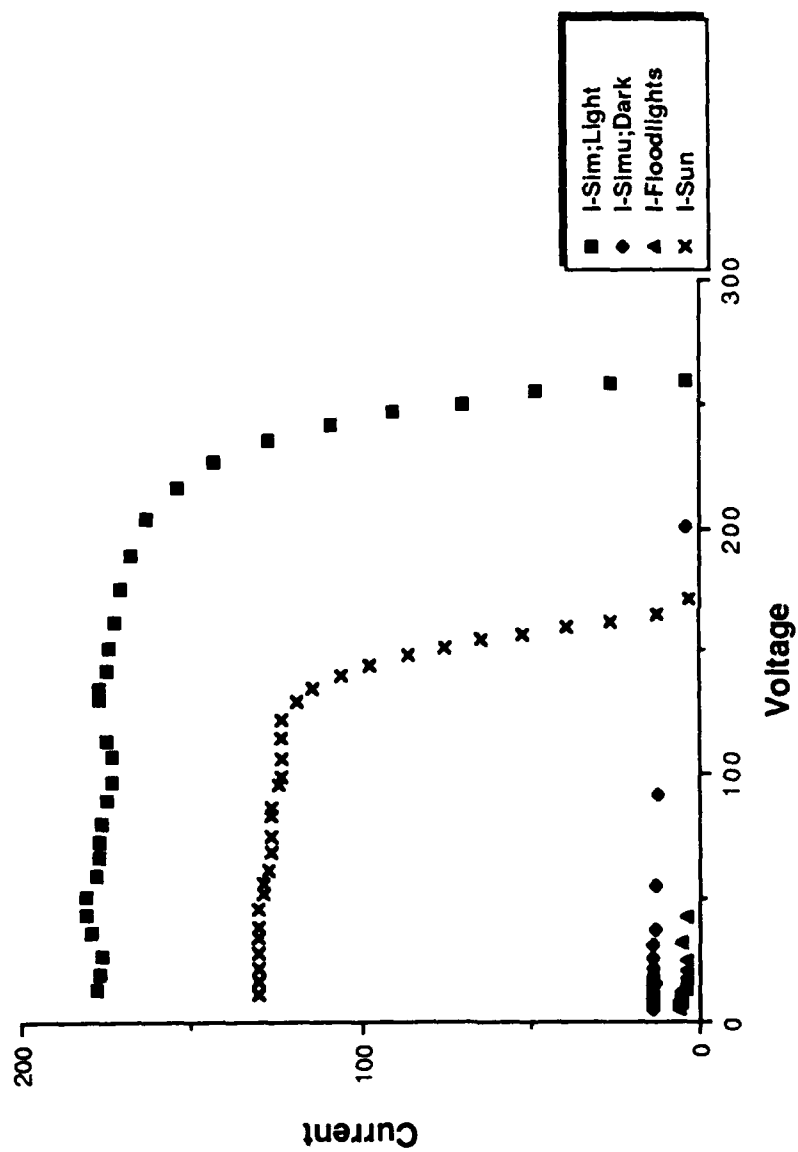


Figure 2.3.1.2.2-2 Mini-Cassegrain Array IV Curves

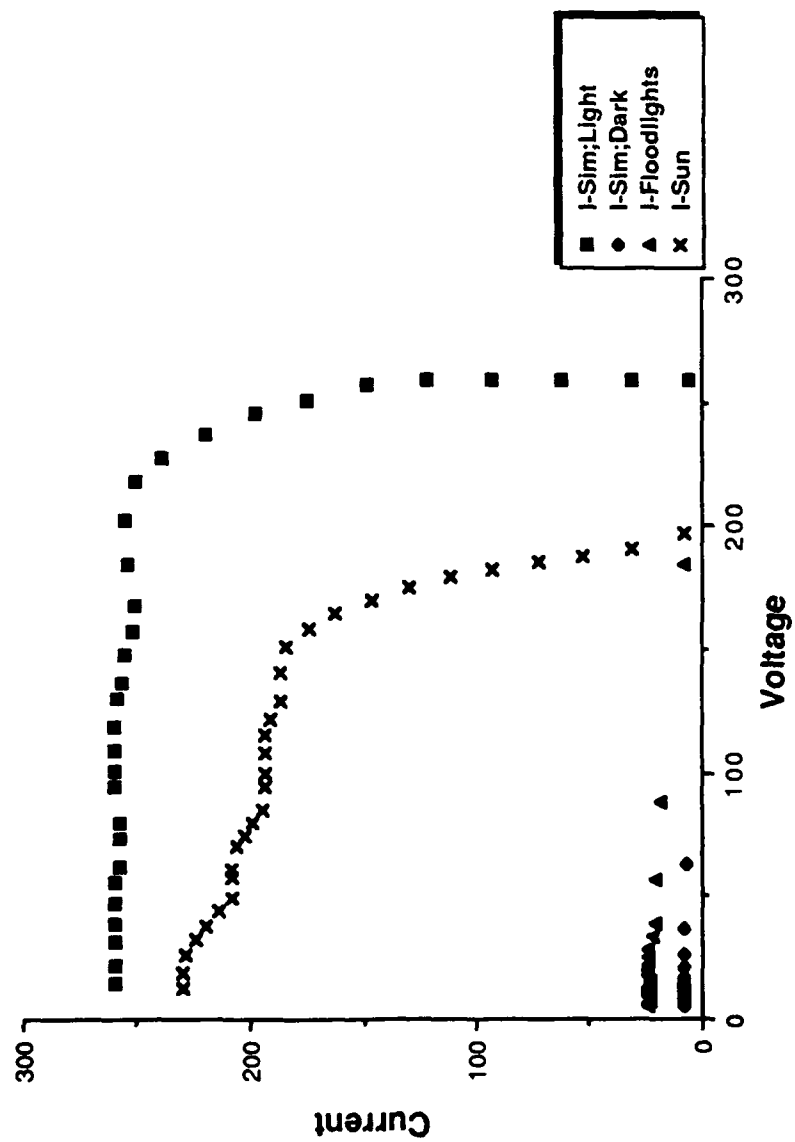


Figure 2.3.1.2.2-3 SLATS Array IV Curves

Each of the IV circuit boards uses a power cutoff relay to conserve power when not in use. This relay is activated by the IVBRD output of the RLL board. The output voltage range for the measured current (ITELEM) and voltage (VTELEM) can be set to fall within the 0-5V range of the A/D convertors in the DACS by adjusting the values of R10, R11, R12 and R13. Similarly the appropriate load levels for an array can be set by selecting the appropriate binary resistance set for R1-3 and R5-9.

2.3.1.2.3 High Voltage Power Supply and Electrometer - The high voltage bias power supply (HVPS) system used in this experiment is derived from the design used in the PIX experiments and the proposed VOLT-A experiment. The function of the HVPS is to supply programmed bias voltages from 0 to plus or minus 500 V in eleven steps.

2.3.1.2.3.1 High Voltage Power Supply Boards - Programmed commands from the DACS control the output of the HVPS. The HVPS is turned on by a HV or HVT2 mode command received by the RLL board.

High Voltage Control Board - The high voltage bias level is set by a five bit command signal from the DACS (HV0-HV4) to opto-isolators U11 and U12 on the high voltage power supply control board (HVC). This digital command is inverted by U14 and then converted to an analog reference signal by a D/A convertor, U1. The output from U1 is a voltage which is proportional to the desired high voltage output. This output is amplified by op amp U2 and buffered by op amp U3. The U4 op amp is used for error correction and then the output (HV DRIVE) is sent to the high voltage driver (HVD) board. A feedback signal is generated from a comparison between the high voltage bias through op amps U5 and U10. This feedback signal and the analog reference voltage are compared in op amp U4.

The current output of the high voltage bias is also sampled by the voltage comparator, U6. If the current goes above 100mA then the retriggeable monostable multivibrator, U8, is triggered. This action creates an over-current cutoff signal, O/C CUTOFF, which is sent to the HVE board and to the DACS through the opto-isolator, U13. The over-current cutoff is set to last for about 2 seconds. A high voltage polarity indicator signal, HV POL IND, is received from the HVE board and also sent to the DACS through U13. The remainder of the HVC board consists of three isolation amplifiers (U16, U17 and U18) which are used to send analog signals to the DACS. The high voltage bias current leakage to the plasma, CTELEM, and the high voltage bias voltage, ETELEM, signals are sent with a gain of one. The electrometer, ELECT, signal from the HVE board is offset and amplified by the isolation amplifier before being sent to the A/D convertor boards of the DACS.

High Voltage Driver Board - The high voltage driver board (HVD) receives the HV DRIVE signal from the HVC board which is then run through an isolation amplifier, U2. The drive signal is then fed

to a pulse width modulator regulator, U3, which drives a push-pull inverter circuit. The inverter circuit converts the input spacecraft +28 V supply to a 2-25 V drive voltage output which is sent to the high voltage oscillator board, HVO. The pulse width modulator also creates a synchronization signal (SYNC) which is also sent to the HVO board to keep its oscillator in step. If there is a need to shut down the HVPS the soft start pin on the U3 is grounded on both the HVO and the HVD boards.

High Voltage Oscillator Board - The 2-25 V drive voltage is delivered to the primary coil center tap of the output transformer on the high voltage oscillator board (HVO). The output transformer, T1, is driven by the pulse width modulator, U1, which is running as an oscillator. Output from T1 is sent to two full wave rectifiers from the split secondary coil of T1. The HV OUT voltage is sent to the high voltage electrometer board (HVE) where it is used to measure array leakage current and can be polarity switched before being sent to the RLR board for application to the appropriate array.

Relay K1 is used to turn on the isolation amplifier power (+15V AD210) for the HVPS boards when commanded by the HV and HVT2 modes signals from the RLL board. Relay K2 is used to ground the soft start pins of the two pulse width modulators if an O/C CUTOFF signal is received from the HVC board.

High Voltage Electrometer Board - The high voltage electrometer (HVE) measures the high voltage bias voltage across resistors R4 and R7. This voltage is then sent to the isolation amplifier U5. Output from U5 is sent to the DACS as ETELEM and to the HVC board for voltage feedback control.

The HVE measures currents from 1 mA down to 100 nA at potentials up to plus or minus 500 volts by splitting the range into high leakage and low leakage channels. Large plasma leakage currents are sensed across resistor R3 and then are sent to the isolation amplifier U6. Output from U6 is sent to the DACS as CTELEM and the HVC board for over-current monitoring. Small leakage current flow to the plasma from an experiment surface is sensed across R1 and then amplified by op amps U3 and U2. The output from the op amps is sensed by a log amplifier circuit which provides an ELECTROMETER OUT signal which is sent to the HVC board. Because the electrometer circuit is tied to the spacecraft ground it must be driven by a separate power supply, +15V HVE.

The polarity of the HVPS output is switched by simply connecting the array leads differently as has been explained in the RLR board section above. The polarity switching is performed by relay K1 using the REV POL signal from the RLL board through the Darlington driver, U1. Another signal is received from the RLL board - the HV ON. This signal also goes through U1 and activates the +15V AD210 power switching relay of the HVO board. Finally, the O/C CUTOFF

signal from the HVC board is used to activate the O/C CUTOFF relay on the HVO board after going through U1.

2.3.2 GFE Solar Array Modules

At present only four solar array modules are available to be tested. Modules 1 (SP) and 2 (GP) are representative of the standard configuration now used for arrays in space and will act as bench-marks for the other test modules and as a basis for comparison with previous flight data from the PIX experiments. Modules 3 (CC) and 4 (SC) represent advanced Air Force concentrator array designs that show high potential for survivability. Both concentrator modules are about 11 x 12 in. while the planar modules are 10 x 20 in. in size. Because each of the modules has a unique configuration, this group of modules is well suited to expand the existing scientific data base on the subject of plasma interactions. Modules 5 and 6 will be either duplicates of the previously described modules or will be other advanced-technology modules provided by AFWRDC at a later date.

Module 1, designated as SP, consists of 100 series connected silicon solar cell assemblies (covered solar cells), 2 x 4 cm in size, mounted on an insulated aluminum honeycomb substrate. Module 2, designated GP, is an assembly of two parallel strings with each string containing 100 series connected gallium arsenide (GaAs) solar cell assemblies, 2 x 2 cm in size. Module 2 is mounted on the same type of substrate as the silicon array module.

Modules 3 and 4 are designated CC and SC respectively. Module 3 is composed of an eight element concentrator using GaAs solar cells. This design, known as the mini-Cassegrainian concentrator, is capable of providing an effective solar concentration of approximately 80 times normal. Module 4 is another concentrator configuration, the survivable low-aperture trough system (SLATS).

2.3.3 Peripheral Sensors and Experiments

The peripheral sensors and experiments collect data on the environment so that the array biasing results can be correlated with environmental conditions.

2.3.3.1 Sun Sensor

The sun sensor used in the PASP Plus experiment is the Adcole system which has flown on numerous space experiments in the past. This digital sun sensor system consists of a sensor head and an electronics package which digitally encodes the sun angle in nominal 0.5 degree increments over a field of view of 128 degrees in each of two orthogonal axes. An automatic threshold adjust (ATA) pattern is used to sense the solar intensity in order to set the current threshold levels in the data bit comparator array. The

A axis and B axis ATA signals are amplified using transconductance amplifiers. The outputs of these amplifiers are then used to set the threshold currents for the data bit comparators which output the two 8 bit parallel data words. This system is available as a shuttle qualified system. In order to obtain accurate IV measurements of the concentrator modules, the modules must be pointed to the sun with an accuracy of at least plus or minus 1.0 degree. The planar modules, however, can tolerate a much higher off-sun pointing error (± 20 degrees). Since sun angle data can be used both by the PASP Plus experiment and by the spacecraft there is a special circuit board in the DACS which provides data to PASP Plus and to a spacecraft interface connector.

2.3.3.2 Langmuir Probe

The key instrument to measure the plasma environment in the vicinity of the solar array modules is a Langmuir probe. The probe is capable of measuring plasma density, electron temperature and the electron distribution function. Due to charging of the spacecraft by the plasma, a potential difference exists between the spacecraft ground and the potential of the space plasma. Since the current collected by the probe depends upon the potential difference, the potential of the spacecraft with respect to the plasma can also be determined. This potential provides very important information on spacecraft charging. For the PASP Plus experiment, the Langmuir probe has a JPL-provided output isolation card in order to maintain proper ground isolation.

2.3.3.3 Transient Pulse Monitor (TPM) Experiment

Two types of electrostatic discharge (ESD) detectors are used by the PASP Plus instrument to measure both radiated and hard wire arc-induced noise. A TPM experiment has been made part of PASP Plus to provide this data. A TPM antenna (TPM S1) is mounted near the modules to detect radiated energy caused by an arc event. A current loop pickup coil (TPM S2) is placed in the electrometer portion of the HVPS to sense the hard wire (conducted) portion of the arc event. All of the electronics for the ESD detectors are located in the TPM electronics case which is located external to the PASP experiment. The PASP experiment is responsible for supplying power and commands to the TPM and incorporating the TPM data into its output data stream by use of a buffer. A special circuit board in the DACS is used as the interface between PASP and the TPM.

2.3.3.4 Pressure Gauge (PG) Experiment

The PG is used with the PASP experiment to determine the partial pressure in the area of the experiment. The PG consists of two separate units - a cold cathode sensing head and an electronics package. The PASP experiment is responsible for supplying power to the PG and incorporating the PG pressure and temperature data into

its output data stream. There are no operating commands required by the PG.

2.3.4 Flight Software

Development of flight software was centered around the restricted amount of on-board memory. FORTH was the chosen language since it lends itself to very sparse coding. A listing of the flight software is found in Appendix H and the attendant memory map is found in Appendix I. The flight software consists of a number of individually defined one, two, four, and six byte data words which are then concatenated into various data marker and event marker sequences to define the data taken or operation for one sensor or instrument during one portion of an orbit. By extending the data marker and event marker sequences a complete orbit is defined and, depending upon the requested activity, a whole test sequence of six orbits (see section 2.4.5) can be performed with one ground command.

2.4 GSE DESCRIPTION

The Ground Support Equipment (GSE) for the Photovoltaic Array Space Power Plus Diagnostics (PASP Plus) experiment provides implementation of a number of services to the PASP Plus Instrument. These services are detailed below after a section on equipment safety and handling. After a description of the services there is a section containing GSE hardware details.

All tests and calibrations are under the control of and are logged by the GSE computer during subsystem and system testing of the PASP Plus instrument. The GSE calibration procedure is briefly described below and in more detail in Appendices D and E. After calibration there is a description of how the PASP Plus experiment may be implemented in the flight mode.

The PASP Plus GSE software is an extension of the PASP Plus flight software contained in the DACS PROMs. This software is described later in this section along with menu and engineering software.

2.4.1 Safety and Handling

There are two safety issues which personnel operating the PASP Plus instrument need to be aware of. They are a high voltage hazard and a GSE rack instability hazard. In addition there are special provisions that must be made when packing or transporting the GSE rack.

2.4.1.1 Safety Hazards

2.4.1.1.1 High Voltage

Inside both the GSE instrument rack and the brassboard PASP

Plus instrument there are high voltages of +/- 500 volts generated by the high voltage bias power supply during certain portions of instrument operation. In addition, the solar array simulators in the GSE rack generate voltages in the 100-125 volt range during certain portions of instrument operation. Similarly the two planar flight solar arrays themselves produce voltages in the 100-125 volt range when illuminated with terrestrial solar irradiance levels. Also the GFE pressure gauge electronics generates and uses high voltage of 2000 volts during operation. Therefore, standard laboratory personnel safety precautions should be exercised when equipment operation is undertaken.

Opening of the GSE rack access door automatically shuts off power to the rack by way of a bypassable interlock switch located on the lower left of the access door. Bypassing the interlock can be accomplished by pulling the plunger out. Care must be used when working in the rack after by-passing the interlock switch. Removal of a GSE chassis, working around the demonstration cart, or disassembly of any portion of the instrument should not be attempted unless all power has been turned off and verified. If more than one person is working with the GSE for maintenance or repair, it would be best if the GSE were unplugged during these activities. In addition, the instrument utilizes a complex system of grounding and ground isolation. Therefore, care should be taken if any ancillary equipment other than that provided as part of the GSE is used or connected to the instrument.

2.4.1.1.2 Rack Instability

The main chassis of the GSE rack are mounted on sliding rails for ease of handling. Due to their weight, care should be exercised when sliding out chassis to avoid an unstable balance condition of the rack. A good rule is to only slide out one rack at a time.

2.4.1.2 Handling Provisions

2.4.1.2.1 When packing, unpacking, or storing the GSE rack, the printer head and floppy disk heads located in the GSE rack should be properly secured and the hard disk head should be parked.

2.4.1.2.2 The computer keyboard is normally stored in the back door of the rack and may be packed separately for shipment.

2.4.1.2.3 There are removable sub-chassis inside the GSE. These items are held in place only with Velcro and must be strapped down or packed separately when the GSE is transported.

2.4.2 GSE Implementation

The PASP GSE is a broad-based system intended for use during system testing as well as during instrument and software development, fabrication, testing, and calibration at the subsystem level. For

example, the GSE contains complete ASIS and DACS instrument systems composed of breadboard and brassboard hardware. This capability allows modular testing and trouble-shooting capability for breadboard and prototype subsystem boards during instrument development and checkout.

2.4.2.1 DACS Subsystem Services

A brassboard DACS is part of the GSE to allow testing of DACS cards and to simulate the flight DACS for software development, subsystem test, and calibrations. The DACS card cage is also used to house special communication circuit boards needed during system development.

2.4.2.2 ASIS Subsystem Services

2.4.2.2.1 High Voltage, Emitter and Electrometer System

A brassboard high voltage power supply (HVPS) is supplied in conjunction with a simulated emitter current path. A HVPS consisting of a control board (HVC), driver board (HVD), oscillator board (HVO), and electrometer board (HVE) can be characterized by substituting the test units for the GSE mounted units. Voltage and current measurements from the ASIS are fed to the DACS for digitizing, buffering and packetizing. The DACS then forwards the data to the GSE computer for storage, display or hard copy printout.

2.4.2.2.2 Relay Switching and Logic

A brassboard relay switching and logic circuit (RLL & RLR) is incorporated to allow testing of circuit cards.

2.4.2.2.3 IV Curve Measurement

Six brassboard IV curve measurement circuit boards (IV1-6) are provided to allow testing of IV circuit cards.

2.4.2.3 GSE Simulators

Simulation of the flight solar array modules is provided by array simulation circuit boards (AR1-6). The array simulators work with computer generated loads to produce an IV curve from open circuit voltage to short circuit current. Array leakage currents and arcing events can also be simulated. Sun sensor position digital data can be simulated and analog signals from the Langmuir probe are also available. A simulator for the TPM and PG sensor outputs is included to provide appropriate test signals for insertion into the data stream.

2.4.3 GSE Hardware Description

The PASP Plus GSE electronics consists of an instrument rack of hardware containing various flight test and IMPS interface simulators; digital readout instrumentation; and an IBM XT-compatible controller to serve as the control center. The controller contains the necessary boards to interface with the GSE test sensor and interface simulators and instrumentation. A functional description of each of the chassis in the GSE instrument rack is given in the following sections. Schematics for all of these chassis have been prepared.

The GSE unit provides the power, experiment simulation, interface simulation, dedicated IBM XT-compatible computer for control and operation, and recording equipment necessary to exercise, operate, and document the PASP instrument performance during fabrication and testing. The GSE consists primarily of two functions; the interface subsystems and the implementation subsystems. Block diagrams of the GSE and cabling to the PASP Plus instrument are available.

There are two groups of interface simulation subsystems in the GSE. The first group of interface subsystems addresses the actual experiment flight hardware test solar arrays and sensors which are not present during almost all of the instrument testing phases of the program. The second group of interface subsystems addresses the carrier interfaces of power input, command and timing input, data output, and control functions which are absent during testing. Since a carrier is not yet defined, these interfaces currently use the old IMPS formats. The interface subsystems are located on various chassis and are described in the section on that chassis.

2.4.3.1 DVM Chassis

At the top of the rack is the DVM chassis which houses two Keithly Model 197 auto-ranging digital multimeters. These instruments are six digit visual readout instruments with IEEE-488 interface buses for communication and data transfer to the controller. Their accuracy is plus/minus one count in the least significant digit. They are used for both manual and automatic data readout and transfer during both calibration and system operations.

2.4.3.2 Calibration Chassis (CAL)

Below the DVM chassis in the rack is the calibration chassis which contains the voltage and current power supplies used to apply known variable inputs to the various portions of the system for calibration of the resulting outputs in the system telemetry data stream. The chassis contains switching capability covering all of the functions of the PASP Plus instrument. The data obtained from system calibration is the primary source of transfer functions for subsequent flight data reduction. Details of the chassis front panel are given below in a left to right sequence.

2.4.3.2.1 Fuse

At the far left of the front panel is the 1/8 amp fuse for the protection of the instrument HV power supply.

2.4.3.2.2 ARRAY/RTD/HV LOAD Selector Switch

This switch selects either an ARRAY input channel, an array RTD channel, or a HV load resistor depending on the setting of the METER FUNCTION selector switch.

2.4.3.2.3 Power Supply Controls

There are two variable power supply setting controls for the 0-20 volts, 20-120 volts, and 0-1 amp sources. The particular type and value of these sources is determined by the settings of the ARRAY/RTD/HV LOAD and METER FUNCTION selector switches.

2.4.3.2.4 RTD SIMULATOR Selector Switch

This switch selects a resistance value to simulate a RTD temperature sensor at a given temperature.

2.4.3.2.5 ELECTROMETER CURRENT Selector Switch

This switch provides selected currents to the electrometer when the METER FUNCTION selector switch is in the ELECT position.

2.4.3.2.6 DMM Coax Outputs

The DMM 1 connector feeds the input calibration signals being applied to the PASP Plus instrument to DMM 1 in parallel. The DMM 2 coax output on the far right simultaneously feeds the instrument output signals to DMM 2.

2.4.3.2.7 EMITTER and AD Power Supply Controls

There are two controls, one for the 0-100 milliamp EMITTER power supply and the other for the -5 to +5 volt AD power supply used to calibrate the AD cards in the DACS.

2.4.3.2.8 METER FUNCTION Selector Switch.

In the IV-V and IV-I positions of the METER FUNCTION switch the IV cards in the ASIS are calibrated for voltage and current respectively using the appropriate power supplies previously described. In the HV-V and HV-I positions the output of the HV power supply in the ASIS is monitored as a function of the load selected by the ARRAY/RTD/HV LOAD selector switch. The ELECT position allows electrometer calibration using the ELECTROMETER CURRENT selector switch. The EMITTER position allows the calibration of the emitter monitor circuit using the EMITTER power supply. The AD position

allows the calibration of the DACS AD cards using the AD power supply.

2.4.3.3 Sensor Simulator Chassis (SIM)

This slide-out chassis contains six array simulators, six HV arc simulators, six RTD simulators, a sun sensor simulator, an emitter simulator, and a Langmuir probe simulator. In addition, all of the necessary isolated power supplies to operate the simulators and readout devices are contained on the chassis. Each of these simulators are covered in turn with a functional description being followed by a panel activity description.

2.4.3.3.1 Solar Array Simulator

The test solar arrays are subjected to two distinctly different types of measurements in a serial manner during the course of the mission. The first measurement is the acquisition of a solar array current-voltage characteristic curve, referred to as an I-V curve. The second measurement consists of biasing an array sequentially over the range ± 500 volts DC and measuring leakage currents and arcing events between the array and the plasma or ground. Two separate solar array simulator circuits are provided to perform these simulations.

2.4.3.3.1.1 I-V Curve Measurement

The GSE contains six unique solar array simulators, each designed to output an I-V curve having the same characteristics as its equivalent flight solar array under one-sun solar illumination during the actual flight. These simulators are used to simulate the maximum current and voltage conditions of the solar array as a source during I-V curve measurements. The simulated I-V curves are obtained and processed by the DACS electronics (just as a real IV curve) then read, displayed, and verified by the GSE.

When the SUN/DARK toggle switch is set to SUN, the array simulators are powered up and the indicator for the selected array simulator will brighten. When an IV command occurs, the selected array lamp will extinguish and then return to its bright state as the IV curve is swept, which occurs in about one second. When the SUN/DARK switch is set to DARK, the array simulators are powered down. It is advisable to power down the array simulators when they are not in use as they consume considerable power and there is an overheating concern.

2.4.3.3.1.2 High Voltage Bias Measurement

During the high voltage bias portion of the test sequence, the solar array simulators described above are switched out and replaced with variable impedance loads to provide bias voltage and leakage current data signals. These signals are processed by the

DACS electronics, and then read and verified by the GSE over the designed operating range of the high voltage power supply and electrometer.

When the high voltage power supply is activated, the HV monitor lamp on the ASIS front panel will begin flashing and the safety precautions previously discussed in the Safety Section should be observed. This warning is especially important when the HV tests are being run on circuits located on the demonstration cart. As the HV bias is applied, the arc tubes can be set to fire thereby simulating an arc on the selected array. Each of the arc tubes is set at a different voltage covering the range of 150-500 volts. The POLARITY SENSE switch determines which side of the array the arc tubes are connected to and a center-off position is provided when operation is not desired.

2.4.3.3.2 Solar Array Temperature Simulator

The Resistance Temperature Detector (RTD) sensor simulators are a set of six actual platinum-resistance temperature detectors. One of the detectors is bonded to the case of the power transistor used to simulate the operation of an array. The temperatures read will be the temperature of the transistor case and will change with time.

2.4.3.3.3 Sun Sensor Electronics Simulator

The sun sensor electronics simulator is a copy of the actual flight sun sensor electronics. The GSE computer sends preprogrammed 16 bit gray coded words representative of predetermined sun pointing angles to the sun sensor electronics simulator input interface to simulate the sun sensor head assembly. The data flow through the sun sensor electronics and DACS electronics is read, decoded, and verified by the GSE as equivalent to the simulated input sun pointing angles. There are two data output interfaces for the PASP sun sensor. The primary interface is a dedicated interface connector through which sun sensor data is provided to IMPS for attitude control throughout the mission. The secondary interface is to the DACS where sun sensor data is incorporated into the data packet to the IMPS AK as part of the secondary header and IV data. The same sun sensor head simulator signal drives both output interfaces.

There are four sun sensor indicators; two readouts for the X and Y axis decimal value of the hexadecimal words being sent by the controller to the sun sensor electronics, and two readouts for the X and Y axis ATA bits indicating presence of the sun within the field of view of the sun sensor head.

2.4.3.3.4 Emitter Simulator

Since the emitter consists of a bare tungsten, hairpin-shaped

filament, operation of the filament is not possible during the normal cycle of ground testing. The emitter simulator is a 5 volt, 25 watt lamp with the same voltage and load values as the actual emitter filament to verify proper current and power drain as supplied by the emitter power supply during those portions of the sequence that the emitter is commanded on. Emitter current and voltage are monitored by the GSE. In addition a separate circuit provides the proper drive voltage and load impedance to simulate filament emission current to test the emission current sensor in the DACS electronics for proper operation. Emission current telemetry readout is verified by the examining the data captured by the GSE.

The emitter indicator lights when the emitter is commanded on and the emitter power supply is loaded by the 25 watt lamp.

2.4.3.3.5 Langmuir Probe Electronics Simulator

A copy of a portion of the Langmuir probe electronics has been constructed to output the anticipated voltage sweep signals of the Langmuir probe. The voltage signals are swept in the same time frame as the actual instrument. This data is then processed by the DACS, placed on the output telemetry lines and read by the GSE computer which acts as a ground station recorder.

The Langmuir probe indicator indicates the presence and amplitude of the sweep voltage being sent to the Langmuir probe head.

2.4.3.4 ASIS Chassis

The ASIS slide-out chassis contains the GSE ASIS test-bed portion of the instrument as well as a removable sub-chassis containing the circuitry which operates the front panel ASIS status monitors. When the sub-chassis is mounted in the GSE ASIS chassis, it monitors the test-bed ASIS. When the sub-chassis is removed and connected to the instrument ASIS it monitors that subsystem through a cable extension that connects back to the ASIS chassis. When the ASIS is in the IV mode, the ARRAY# monitor indicates which of the six arrays is being measured. The IV SAMPLE monitor, a hexadecimal readout, indicates which of the 32 IV curve samples is currently being read. Since the IV curve is swept in about one second, this readout will normally be a blur; however, in the event of a system hangup, the monitor will indicate where the sweep stopped. The HV LEVEL monitor, a decimal readout, indicates which of the 32 available HV power supply voltage levels is commanded on over the range of -500 to +500 volts. The ASIS MODE monitor reads "1" for the IV mode and "2" for the HV bias mode. In the case of the concentrator arrays the conventional HV bias mode is run with the concentrator frames isolated from spacecraft ground. However, there is a third mode for the concentrator arrays (the HVT2 mode) in which the frames are tied to the negative side of the array while the bias is applied to the positive side of the array. When

this mode is commanded, the monitor will read "3". The HIGH VOLTAGE monitor indicates the actual high voltage being generated by the power supply during the stepped bias sweep. The RSL POWER monitor indicates that the ASIS chassis is powered. The HV POWER flashing monitor indicates that the HV power supply is powered and capable of producing high voltage. This flashing indicator, when on, requires that operating personnel exercise high voltage safety requirements and procedures.

The ASIS ambient pressure sensor simulator is also mounted on this chassis. This pressure simulator just produces a DC analog signal between 0 and 5 V.

2.4.3.5 Controller Chassis

The controller slide-out chassis contains an IBM XT-compatible mother board, two 5 1/4 inch 360K floppy disk drives, a 20 megabyte hard disk, CRT, power supply, and the dedicated boards built to run the GSE and the PASP Plus instrument. The rack mounted controller was installed for the purpose of having a GSE dedicated integral controller that was rugged and small enough to meet the needs of the GSE. The floppy disk drives or the hard disk are used for both the loading of control software and the capture of data files from the data stream as the PASP Plus instruments operates.

2.4.3.5.1 IMPS Interface Simulators

The PASP Plus/IMPS interfaces consist of a power connector, a command, timing and data connector, and a sun sensor data connector. These interface connectors communicate, on the IMPS side of the interface, with two elements of the Data Handling Subsystem (DHS); the SPAS Modular Digital Universal System (MODUS) and the Adaption Kit (AK) System. The AK System in turn contains both the AK Data System and the AK Power Distribution Unit (PDU). The GSE contains both hardware and software to simulate these interfaces.

2.4.3.5.2 IMPS Adapter Kit Data System Simulator

This simulator is used to simulate the AK Data System hardware and software as described in the IMPS Interface Control Document. Preprogrammed data are provided as instrument input and instrument output is monitored during instrument testing. The command, timing, and telemetry interfaces of the simulator conform to the specifications contained in the IMPS ICD. Telemetry data consists of both real time data output and tape recorder data output for appropriate distribution.

2.4.3.5.3 IMPS MODUS Simulator

This simulator is used to simulate the MODUS hardware and software as described in the IMPS ICD. Preprogrammed commands are provided as instrument input and instrument output is monitored during

instrument testing. The command, timing, and telemetry interfaces of the simulator conform to the specifications contained in the ICD. Telemetry data consists of both real time data output and tape recorder data output for appropriate distribution.

2.4.3.5.4 IMPS Ground Support Equipment Simulator

Originally the IMPS GSE interface consisted of an Ethernet system interface. Since the PASP instrument had no hard line stimulus interface at the system level when operating through Ethernet, the PASP GSE interface with the IMPS GSE required only that incoming data be decoded and recorded through the Ethernet interface. Details of the carrier interface are TBD at this time.

2.4.3.5.5 Sun Sensor Interface Simulator

In addition to the inclusion of the sun sensor data in the normal data stream during PASP instrument operation, the IMPS project requested an independent sun sensor data output line throughout the course of the mission. For this purpose a separate, dedicated interface connector has been added to the system with the interface isolation provided by the PASP instrument. A simulated interface circuit is used to monitor sun sensor data signals across the interface and verify operation.

2.4.3.6 TPM/PG Chassis

The TPM/PG chassis contains the TPM and PG simulators. The TPM simulator is controlled by the DACS. The PG simulator consists of two switches on the front panel for selecting the HI/LOW output signals of the PG PRESSURE channel and the PG TEMP channel.

2.4.3.6.1 TPM Simulator

The TPM simulator is an actual TPM printed circuit board with the command capability of the instrument and simulated sensor inputs into the actual flight pulse diagnostic electronics and outputs. The software PROMS in the simulator are identical to the flight unit. The operation of the TPM is controlled by the DACS and the GSE determines the presence of TPM data in the instrument data stream.

2.4.3.6.2 PG Simulator

The outputs of the PG are two 0-5 volt analog telemetry channels for the ambient pressure and instrument temperature. The simulator provides two output channels, each with a switchable HI and LO output. The temperature channel outputs are 4 and 2 volts and the pressure channel outputs are 3 and 1 volts. These signals are fed to the DACS and read by the GSE to verify operation through the PASP Plus instrument.

2.4.3.7 DACS Chassis

The DACS slide-out chassis contains the GSE DACS card cage test-bed portion of the instrument. The DACS chassis also contains the LPE and SSE simulator electronics in removable sub-chassis. Both of these electronics are hooked to the Simulator Chassis for readout purposes and have already been described above in that section. The DACS ambient pressure sensor simulator is on this chassis and operates identically to the ASIS pressure sensor simulator. The ASIS manual control unit (AMCU) can also be located on the DACS chassis.

2.4.3.8 Power Control Chassis

The power control chassis accepts the main 117 VAC input and distributes it to appropriate chassis in the GSE rack. This chassis also contains a separate filtered switchable distribution system for the controller and its peripherals.

2.4.3.9 Keyboard Chassis

A slide-out tray is provided for placement of the controller keyboard for use during operation of the GSE. The keyboard plugs into the power control chassis immediately above it. Due to the lack of space in the GSE rack, the keyboard is stored separately in the back door of the GSE rack.

2.4.3.10 Printer Chassis

The slide-out printer chassis contains an Okidata Microline 320 nine pin dot matrix printer capable of up to 300 characters per second. This printer is used when hard copy output of software or data is desired.

2.4.3.11 Power Simulator Chassis

The power simulator chassis contains the main 28 VDC power supply which simulates the main power bus of the carrier to the PASP Plus instrument in flight configuration. The chassis contains current and voltage meters for the determination of instrument power consumption during operation.

This 28 VDC $\pm 4/-3$ VDC, 10 amp power supply powers the PASP Plus instrument during subsystem and system testing and has a capacity of at least double the absolute maximum peak power required by the PASP Plus instrument. While this power supply allows operation of the instrument over the DC voltage range specified in the ICD, it does not have the capability of simulating the ripple and transient voltage excursions specified in the ICD.

2.4.4 GSE Calibration Procedure

The calibration procedure consists of generating a voltage or current signal in the Calibration Chassis and routing this signal through the appropriate data channel to the DACS. The DVM Chassis monitors both the applied signal and the resultant signal after the DACS. These before and after values then define a transfer function for the data channel. These transfer function values can be captured during the calibration process in transfer function files on a floppy disk in the Controller Chassis. These files can then be used to calculate actual data using transmitted data from the various instruments and sensors. Calibration curves from the instruments and sensors are also required and have been generated during the PASP program for the sun sensor and the GFE arrays. A much more detailed description of the calibration procedure can be found in the System Test Report in Appendix E.

2.4.5 Flight Operation Mode

During flight operations the PASP experiment may be commanded to perform any one of six different orbit test sequences or allowed to run free through the entire set of six orbit test sequences. A low power mode and a standby mode for PASP can be commanded as can a warm restart in case of system hangup. A more detailed description of available flight commands can be found in the PASP Plus System Outline in Appendix A.

The complete flight sequence requires six orbits to complete. These six orbits are not required to be contiguous since each of the six orbit sequences can be commanded ON individually, but in the interest of conservation of orientation propellant, it would be desirable to conduct the orbits consecutively. The experiment sequence for each orbit is unique in terms of which four of the six arrays are tested, whether the concentrator array frame is grounded or not, and the state of the emitter (on or off). The pre-programmed test arrays and conditions for each orbit sequence are shown in Table 2.4.5-1 below.

Table 2.4.5-1 - Module Test Matrix

Orbit Number	SP Array	GP Array	CC Array	SC Array	TBD Array	TBD Array	Emitter State
1	X		X	X		X	OFF
2	X		X	X		X	ON
3			O	O	X	O	ON
4	XX	X			X		ON
5			O	O	X	O	OFF
6		X	X	X		X	OFF

Notes: X = Module selected for normal test

XX = Module selected twice in this orbit for normal test

O = Concentrator frame connected to array negative lead

Apart from the selection criteria shown in the table, the orbit sequence for all six orbits is identical. An eighty-four minute orbit sequence consists of three identical data acquisition elements (DAE) each fifteen minutes, thirty-nine seconds in length during the sunlit portion of the orbit (see Table 2.4.5-2 below for details) and an occultation element of thirty-four minutes, twenty-five seconds during the dark portion of the orbit. Note that the remaining time of the 83 minute, 52 second orbit is taken up by two quiet periods between the three DAEs (about one minute, fifteen seconds each). An orbit is defined to begin when the spacecraft leaves occultation.

The DAEs allow data to be collected in the north and south polar regions and at the equator. The DAEs are broken down into nine periods as shown below:

Table 2.4.5-2 - Data Acquisition Element

IV 1-6 LP	Bias 1	IV 1-6 LP	Bias 2	IV 1-6 LP	Bias 3	IV 1-6 LP	Bias 4	IV 1-6 LP
--------------	-----------	--------------	-----------	--------------	-----------	--------------	-----------	--------------

In the IV 1-6/LP periods the IV curves for all six arrays are measured which takes about one second per array and then the Langmuir Probe is swept which takes about two seconds. The bias periods are dedicated to the four selected arrays for the orbit sequence as detailed in Table 2.4.5-1 above. The Bias 1 period of

orbit 1 would then be dedicated to the SP array while the Bias 2 period would be dedicated to the CC array. Each bias period consists of eleven high voltage bias steps of twenty seconds each over the range of zero to +/-500 V as detailed in Figure 2.4.5-1.

The ideal flight orbit sequence is described below:

1. Verify sun acquisition (this was not incorporated since no spacecraft interface data was available).
2. Scan house keeping data (power supply voltages, equipment temperatures, enclosure pressures, in-flight calibrations, and error status counts).
3. Measure sun pointing angle, then module temperature and module IV curve for all six arrays.
4. Sweep Langmuir probe.
5. Establish test array number and emitter state according to the selection table.
6. Run first bias voltage sweep. A bias voltage sweep consists of eleven voltage steps for twenty seconds each in the following order: 0, +50, +100, +150, +300, +500, 0, -150, -200, -300, -500. Electrometer current, bias voltage and supply current is recorded each second during the sweep. Emitter emission current is recorded at the end of each twenty second sweep. An emitter OFF command is given after the command to step from +500 V to 0 V (this was not completed during software development). There is a Langmuir probe sweep at the end of the zero voltage period just before the negative bias voltage steps.
7. Repeat steps 2 through 5.
8. Run bias voltage sweep of second test array.
9. Repeat steps 2 through 5.
10. Run bias voltage sweep of third test array.
11. Repeat steps 2 through 5.
12. Run bias voltage sweep of fourth test array.
13. Repeat steps 2 through 4.
14. Pause to the beginning of the next data acquisition element.
15. Run second data acquisition element by repeating steps 2 through 14.
16. Run third data acquisition element by repeating steps 2 through 13.
17. Run occultation element. During the occultation element there will be no bias sweeping or IV measurements. A 20 second data scan will be performed once every 20 seconds. During the data scan the silicon planar array (SP) will be biased to -200 V. The data acquired during the scan will be housekeeping, Langmuir probe sweep, solar array temperatures, electrometer current, bias voltage and supply current. The sun sensor output will be monitored to detect the onset of the sun at the end of occultation to terminate the occultation element (not incorporated

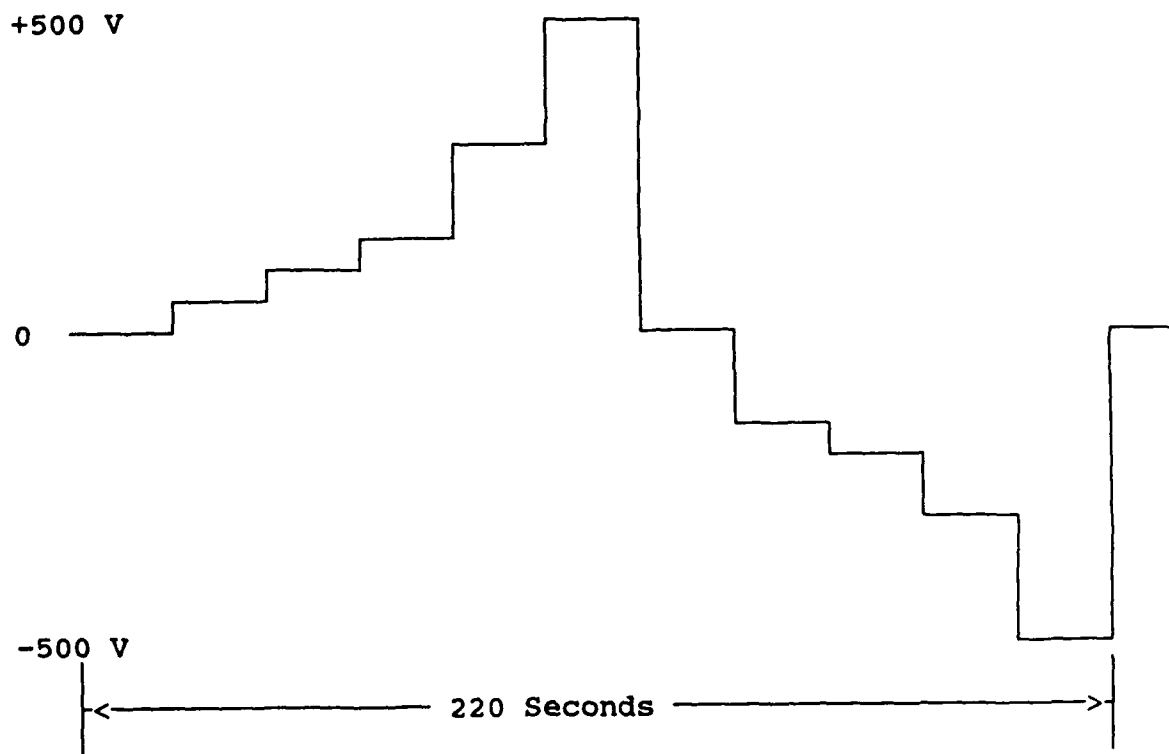


Figure 2.4.5-1 HV Bias Subelement

due to lack of spacecraft data).

During the performance of the orbital sequences, which are controlled by on-board PROMs, the resultant data must be buffered, packetized and then transmitted to the spacecraft. All of these activities are handled by use of on-board RAMs and the flight software stored on the PROMs.

2.4.6 Operating and Development Software

Software development had to extend beyond the narrow needs of the flight data acquisition sequences which form the heart of the flight software. The GSE is required to be able to simulate flight operations during the instrument development period. The GSE can also be used as a command ground station for flight operations if it is suitably hooked into the communications system. These requirements resulted in a large amount of GSE specific software. Also required during development was the calculation of some necessary engineering data. This engineering software is also discussed below.

2.4.6.1 GSE Operating Software

The instrument and simulator hardware is operated by the GSE controller using operating software developed using the F-PC (November 1989 release) version of the FORTH-83 Standard language (Reference: FORTH-83 Standard, A Publication of the FORTH Standards Team, Mountain View Press, August 1983). The operating software performs the functions of operating the instrument, encoding and decoding command sequences, running in-flight calibrations, and placing data outputs on a magnetic record. A series of pull-down menus is provided to assist the GSE operator in selecting appropriate options and operating sequences. Details of the operating commands are found in the System Outline (Appendix A). A listing of GSE menu software is provided as Appendix K and the attendant data dictionary is included as Appendix L. The program source code is included as part of the magnetic record documentation of the project.

2.4.6.2 Engineering Software

During the course of the PASP Plus development effort there was a need to develop some engineering programs. The engineering programs allow calculation of orbit data collection periods, IV curve load commands, and sun sensor gray code words. These programs were developed in BASICA (IBM's advanced BASIC) and are detailed and listed in Appendix F. The BASIC programs are also included as part of the magnetic records of the project.

The IV curve loads program (IVA5.BAS) produces hardcopy and graphical outputs which were very helpful in determining the IV curve simulation hardware. Figures that show calculated IV curves

at temperature extremes based upon actual array physical parameters are attached to section 3.1 of Appendix F.

There are both tabular and graphical outputs from the ARRANAL program. Plots of both short circuit current and peak power for the mini-Cassegrainian and SLATS arrays are located in section 3.2 of Appendix F while a plot of the orbital DAEs (made by ORBITPP) is shown in section 3.3.

A table (3.4-1) attached to Appendix F of this report) contains the hexadecimal and gray code words for the sun sensor and can be calculated by use of the program, GRAY.BAS.

2.5 Sun Sensor Acceptance Test

An acceptance test report for the Adcole Sun Sensor was completed at JPL on 15 June 1988. A letter report was submitted along with an attached test procedure (JPL Procedure No. TP515086, Rev. B). The sun sensor successfully met all of the acceptance test requirements.

2.6 Array Acceptance Tests

Array acceptance testing was performed in two cycles - an initial set of acceptance tests and a secondary set of tests on two arrays which had to be repaired.

2.6.1 Initial Array Testing

Initial array acceptance testing was completed at JPL on 31 October 1988 with the publication of the Final Solar Array Acceptance Test Report. Summary and conclusions from this report are given below in an abridged version.

2.6.1.1 Planar Silicon Solar Array

This array would be usable as a flight array with the addition of adequate cable tie down hardware and a ground strap tie point.

2.6.1.2 Planar Gallium Arsenide Solar Array

This array would require the same additions mentioned above for the silicon array as well as repair of an arc burn area which was formed when the array shorted out during isolation testing.

2.6.1.3 Planar ICA Array

This array would be unusable as a flight array in its present state inasmuch as the objective of the experiment has been compromised by the exposure of a significant amount of cell surface area due to insulation delamination.

2.6.1.4 Mini-Cassegrainian Concentrator

This array would be usable as a flight array in its present state. The data analysis clearly indicates that the angular incidence performance is adequate.

2.6.1.5 SLATS Concentrator

This array would be unusable in its present state within the confines of the original experimental flight plan. The short between the array and the concentrator frame would preclude the application of bias voltage to the array with the concentrator frame grounded. In addition, a non-shortening arcing site would also have to be located and repaired.

2.6.2 Array Retesting

After discussion with the Air Force, it was decided to drop the ICA array from the PASP Plus program. Repairs were made to the GP and SLATS arrays and these two arrays were retested. Details of the retesting are contained in an Addendum to the original Array Acceptance Test Report as noted below. The original acceptance test report has been revised as Revision A on 27 July 1989 by adding Sections 3.2.3, 3.5.3, and 5.6. The gallium arsenide planar solar array and the SLATS concentrator solar array both passed the hi-pot isolation acceptance test after repairs were made.

2.7 SPICE Mission Cost and Feasibility Study

Near the end of the PASP Plus effort, AFGL requested that a cost and feasibility study be made on placing the PASP Plus Instrument aboard a Pegasus expendable launch vehicle. This mission is described briefly below as an indication of the capabilities of the PASP Plus Instrument.

The Survivable Power Interactions in the Charging Environment (SPICE) mission had as its primary purpose the testing of advanced, survivable solar arrays in space. Space testing of prototype array designs is required to provide the understanding needed to design reliable, high-power arrays. SPICE would place a flight version of the Jet Propulsion Laboratory's (JPL's) Photovoltaic Array Space Power Plus Diagnostics (PASP Plus) experiment on a Pegasus launch vehicle. The Pegasus launch vehicle and the spacecraft bus would be built by Orbital Sciences Corporation. It was proposed that the PASP Plus experiment would be built by JPL.

SPICE is really a dual mission. The primary mission is to obtain data on solar arrays in a plasma environment identical to the PASP mission on IMPS. This part of the SPICE mission should only require two to four weeks of data collection to complete. The secondary mission is to obtain data on radiation degradation of advanced, survivable solar arrays. Radiation degradation is of

interest to SDIO and other potential sponsors. This part of the mission may have to run at least one year and possibly as long as two years to obtain sufficient degradation data. Since the original PASP design was to operate for six days, operation for the entire two year mission cannot be guaranteed. Operating for two years in a radiation environment also presents some reliability problems.

The major disadvantage of the SPICE mission as compared to the IMPS mission is the inability to recover the solar arrays for laboratory examination of arcing damage. The major advantages of the SPICE mission is the low cost of the Pegasus launch vehicle and the availability of the PASP Plus design. The Cost and Feasibility Study for the SPICE mission was provided separately and is not included in the Appendices for this report, however, the weight table is included as Appendix J.

2.8 System Tests and Status

System testing was performed according to requirements outlined in the System Test Procedure (Appendix D) and reported in detail in the System Test Report (Appendix E). The system test procedures that were used differ somewhat from the initial IMPS functional and calibration test requirements.

2.8.1 System Test Results

The detailed system test results are given in the System Test Report (Appendix E). All of the requested system tests were successfully performed except the Pressure Gauge (PG) tests and the system thermal tests. The PG tests could not be done since the instrument was returned to the Geophysics Laboratory at their request. After discussion with the Air Force, system thermal tests were dropped for reasons of cost and schedule.

2.8.2 Hardware Status

System hardware was adjusted and necessary small repairs were made during the early portion of the system test program. All major system hardware worked properly and consistently during the final portion of the system test program.

2.8.2.1 Hardware Liens

There are some system hardware deficiencies that should be noted for later consideration and action. These hardware liens are detailed below.

2.8.2.1.1 The GSE controller hard disk does not have an operating light on the front panel. This lack can cause some operator concerns during lengthy file reading or writing operations.

2.8.2.1.2 Resistor values on both sets of the IV4 board current (ITELEM) channel and both sets of the IV1 board voltage (VTELEM) channel could be changed to better match the array outputs. These changes were not made since the arrays were going to be changed at a later date.

2.8.2.1.3 The emitter development was never completed since the emitter boom length is dependent upon the spacecraft geometry. The emitter booms sent with the equipment will not pass vibration requirements.

2.8.2.1.4 The arc lamp circuits were intended to be bipolar with the polarity controlled by a front panel switch. The actual circuit was developed late and is not bipolar. To achieve bipolar operation an arc lamp can be switched on during the appropriate portion of the high voltage test sequence. The arc lamps also do not cause over-current cutoff in the high voltage power supply since the duration of their current spike is only a few microseconds while the over-current cut-off circuit reaction time is over 100 milliseconds. Therefore, the arc lamps should be turned off manually after they have arced a couple of times. Some bad data blocks were noted during arc lamp activation, so they may be coupling excess noise into the system.

2.8.2.1.5 The breadboarded sun sensor interface circuit has the 4 and 8 bit data channels swapped which results in some extra data reduction requirements. The sun sensor simulator Y-axis display used to have the two nibbles inverted; however, this has been fixed.

2.8.2.2 Hardware Enhancements

There were some areas in the PASP Plus program where the development effort resulted in hardware capabilities in excess of those spelled out in the requirements. These areas are detailed below.

2.8.2.2.1 The GSE is a major improvement over a simple bench test apparatus. The GSE has a full set of ASIS and DACS subsystem electronics which are functionally and mechanically identical to the brassboard electronics. This allows board by board or subsystem level testing by replacement thereby speeding development efforts.

2.8.2.2.2 The hardware design was never completed due to the lack of a carrier. However, the design was always kept compatible with later packaging requirements since the printed circuit boards were sized to fit into two cold plate/card cage packages.

2.8.3 Software Status

System software development was a major problem during the last part of the program. A need to create many more software operation

and development tools such as windows and menus was perceived since the software would need to be extensively modified after delivery. These new software development tools are provided along with an improved version of the FORTH language. Instead of the F83 version of the FORTH-83 Standard, the software now used is F-PC, a new version of the FORTH-83 standard which was issued in November of 1989. A number of small modifications were required to upgrade the old program to the new software but this effort was more than offset by the availability of numerous development tools in the new language version.

The down side to this improved software decision was the inability to fully debug all of the planned software attributes. A calibration menu is provided but only a few of the menu choices can be used. The operations and setup menus are fully functional and allow full normal use of the instrument.

2.8.3.1 Software Liens

2.8.3.1.1 All of the orbit sequence commands sequence properly, however, at the end of ORBIT 6 there was no clear indication of the end of a sequence.

2.8.3.1.2 The ETELEM data contained a few bad data points which may indicate the sequence timing was too fast for proper test results.

2.8.3.1.3 The RTD data at the beginning of the IV curve sequence was bad in all cases and indicates a need to lengthen the sequence time between turning on the IV board and reading the data.

2.8.3.2 Software Enhancements

2.8.3.2.1 The need to incorporate the extra commands for the TPM led to a system capability to select up to 16 commands and 16 parameters. This ability means that in-flight channel calibration and noise tests can be performed along with system test sequences.

2.8.3.2.2 The delivered software has programs which can be used for meta compiling of new software and the burning of new PROMs.

2.9 Pre-Ship Efforts

2.9.1 Pre-Ship Demonstrations

Prior to shipment the PASP Plus instrument was demonstrated four times. There was a preliminary demonstration on Tuesday, 24 July 1990, which was given for the JPL TAP Program Office. A more formal demonstration was given on Thursday, 26 July 1990, for JPL management. The demonstration on Wednesday, 29 August 1990, was for the benefit of the Air Force sponsors, the Geophysics Laboratory and the Wright-Patterson Research and Development Center. The final demonstration on Thursday, 30 August 1990, was for the

benefit of the Aerospace Corporation and the Air Force Space System Command. The instrument worked flawlessly during all four demonstrations.

2.9.2 Pre-Ship Review

A formal pre-ship review of the PASP Plus Task was held on Wednesday, 8 August 1990. At that time it was determined that the PASP Plus instrument had met the functional requirements and that the requisite paperwork had been completed. One paperwork issue was set aside. This was the problem with GFE equipment arriving at JPL without adequate paperwork. The outstanding Inspection Reports were closed on these pieces of equipment by noting that they were to be returned to the sponsors in an as-is condition.

The pre-ship review board requested that the PASP Plus hardware be tested in full sunlight prior to shipment. The equipment was subsequently moved outside and a full sunlight test was successfully performed.

2.10 Delivery

The PASP Plus equipment was shipped to AFGL in three wooden crates on 1 October 1990. The GSE went in the largest crate. The PASP Plus brassboard went in another crate along with cables, components, drawings and manuals. The final crate contained the TPM instrument and was the same crate used to ship the TPM to JPL.

A separate shipment of the four usable solar array modules was made to the Wright Research and Development Center in a single wooden crate.

APPENDIX A - SYSTEM OUTLINE

- 1 Objectives
 - 1.1 Provide an outline of system parameters
 - 1.2 Provide an outline of target JPL test capabilities
- 2 System Hardware
 - 2.1 ASIS sub-system
 - 2.1.1 Logic board for ASIS control (RLL)
 - 2.1.2 Relay board for array selection (RLR)
 - 2.1.3 Six IV load and data boards (IV1-6)
 - 2.1.4 Four high voltage power supply boards
 - 2.1.4.1 High voltage driver (HVD)
 - 2.1.4.2 High voltage oscillator (HVO)
 - 2.1.4.3 Electrometer (HVE)
 - 2.1.4.4 Control (HVC)
 - 2.1.5 Auxiliary power supply board (APS)
 - 2.1.6 Housekeeping board (HK1)
 - 2.2 DACS sub-system
 - 2.2.1 Sequence board (SEQ)
 - 2.2.2 *Command interpretation board (CMD)
 - 2.2.3 Resistance temperature device board (RTD)
 - 2.2.4 Two A/D converter boards (AD1-2)
 - 2.2.5 *Mission elapsed time board (MET)
 - 2.2.6 *Real time data output board (RDO)
 - 2.2.7 *Tape data output board (TDO)
 - 2.2.8 *Sun sensor interface board (SSI)
 - 2.2.9 Digital signal board (DIG)
 - 2.2.10 Transient pulse monitor interface board (TPI)
 - 2.2.11 Bus termination board (TER)
 - 2.2.12 *Switched power supply board (SPS)
 - 2.2.13 *Housekeeping board (HK2)

*These boards are interface breadboards
*These boards will be packaged with the ASIS boards
 - 2.3 Emitter (EM)
 - 2.4 Housekeeping pressure sensors (PS1-2, PS2 is simulated)
 - 2.5 Housekeeping temperature sensors (TM1-4)
 - 2.6 Sun sensor
 - 2.6.1 Sun sensor head (SS)
 - 2.6.2 Sun sensor electronics (SSE)
 - 2.7 GSE sub-system
 - 2.7.1 Digital multimeter chassis (DVM)
 - 2.7.2 Calibration chassis (CAL)
 - 2.7.3 Sensor simulator chassis (SIM)
 - 2.7.4 ASIS sub-system chassis
 - 2.7.5 Controller chassis
 - 2.7.6 TPM/PG chassis
 - 2.7.7 DACS sub-system chassis
 - 2.7.8 Power control chassis
 - 2.7.9 Keyboard chassis
 - 2.7.10 Printer chassis
 - 2.7.11 Power simulator chassis
 - 2.7.12 Electronics rack for above chassis

- 2.8 Government furnished equipment
 - 2.8.1 Test solar arrays
 - 2.8.1.1 Silicon planar array (SP, directed buy)
 - 2.8.1.2 Gallium arsenide planar array (GP)
 - 2.8.1.3 Mini-Cassegrainian concen. array (CC)
 - 2.8.1.4 SLATS concentrator array (SC)
 - 2.8.2 Langmuir probe (LP)
 - 2.8.2.1 Langmuir probe sensor head (unmounted)
 - 2.8.2.2 Langmuir probe electronics (LPE)
 - 2.8.2.3 Langmuir probe head simulator
 - 2.8.3 Pressure gauge (PG)
 - 2.8.3.1 Pressure gauge sensor head (PG)
 - 2.8.3.2 Pressure gauge electronics (PGE)
 - 2.8.4 Transient pulse monitor (TPM)
 - 2.8.4.1 Current pulse sensor (TPM S2, mounted on HVE board)
 - 2.8.4.2 Radiated pulse antenna (TPM S1, to be mounted near arrays)
 - 2.8.4.3 Transient pulse monitor elect. (TPM)
 - 2.8.4.4 Antenna pulse stimulator electronics
 - 2.8.4.5 Antenna pulse stimulator head
- 3 Peripheral Equipment Data Outputs
 - 3.1 Test solar arrays (analog signals)
 - 3.1.1 Array current output versus load at 32 points
 - 3.1.2 Array voltage output versus load at 32 points
 - 3.1.3 Array temperature (platinum RTD resistance)
 - 3.1.4 Array DC leakage versus bias voltage
 - 3.2 Sun sensor (2 byte digital signals)
 - 3.2.1 X-axis sun angle ("A" Reticle output)
 - 3.2.2 Y-axis sun angle ("B" Reticle output)
 - 3.3 Langmuir Probe (analog signals)
 - 3.3.1 Sweep voltage (V)
 - 3.3.2 Electron density (N_e)
 - 3.3.3 Electron temperature (T)
 - 3.4 Pressure Gauge (analog signals)
 - 3.4.1 Ambient pressure
 - 3.4.2 Instrument temperature
 - 3.5 Transient Pulse Monitor (44 byte digital signals, see table 3.5-1 below)
 - 3.5.1 Arcing event pulse characterization (6 channels, only 2 used)
 - 3.5.1.1 Positive pulse amplitude
 - 3.5.1.2 Negative pulse amplitude
 - 3.5.1.3 Positive pulse derivative
 - 3.5.1.4 Negative pulse derivative
 - 3.5.1.5 Integral
 - 3.5.2 Additional data
 - 3.5.2.1 Sequence count
 - 3.5.2.2 Pulse count register
 - 3.5.2.3 Software status
 - 3.5.2.4 Threshold gain status
 - 3.5.2.5 Command status

- 3.5.2.6 Parity
 3.5.3 Housekeeping data
 3.5.3.1 Regulated power supply voltage
 3.5.3.2 Electronics box temperature

Table 3.5-1 - TPM Packet Structure

First Packet			Second Packet									
MSB		LSB	MSB		LSB							
1	Sequence Count		0	23	Sequence Count				1			
2	PAMP0			24	PAMP3							
3	NAMP0			25	NAMP3							
4	PDER0			26	PDER3							
5	NDER0			27	NDER3							
6	INT0			28	INT3							
7	PAMP2			29	PAMP4							
8	NAMP2			30	NAMP4							
9	PDER2			31	PDER4							
10	NDER2			32	NDER4							
11	INT2			33	INT4							
12	PAMP1			34	PAMP5							
13	NAMP1			35	NAMP5							
14	PDER1			36	PDER5							
15	NDER1			37	NDER5							
16	INT1			38	INT5							
17	Count0	Count1		39	Thrsh3		Thrsh2		Thrsh1		Thrsh0	
18	Count2	Count3		40	Thrsh5		Thrsh4		Command			
19	Count4	Count5		41	PA0	NA0	PD0	ND0	I0	PA1	NA1	PD1
20	PULSE CYCLES			42	ND1	I1	PA2	NA2	PD2	ND2	I2	PA3
21	VSUP			43	NA3	PD3	ND3	I3	PA4	NA4	PD4	ND4

22	TEMP	44	I4	PA5	NA5	PD5	ND5	I5		
----	------	----	----	-----	-----	-----	-----	----	--	--

Key: PAMPO = Positive pulse amplitude of channel 0
 NAMPO = Negative pulse amplitude of channel 0
 PDERO = Positive pulse derivative
 NDERO = Negative pulse derivative
 INTO = Integral
 Count0 = Pulse register counts
 PULSE CYCLES = Software status
 VSUP = Supply voltage
 TEMP = Box temperature
 Thrsh0 = Threshold status
 Command = Command status
 PA0 = Parity of positive pulse amplitude of channel 0

Reference: Transient Pulse Monitor, Science Report No. 1 (AFGL-TR-88-0147), dated 20 May 1988.

- 4 PASP Plus Command and Control
 4.1 PASP Internal Commands
 4.1.1 System input commands (12 bit digital signal, see table 4.1-1 below)

Table 4.1-1 - PASP Plus System Commands

Command Number	Command Format	Command Description
0		SPARE 1
1	1DE	ORBIT 1 SEQUENCE
2	2ED	ORBIT 2 SEQUENCE
3	3FC	ORBIT 3 SEQUENCE
4	48B	ORBIT 4 SEQUENCE
5	59A	ORBIT 5 SEQUENCE
6	6A9	ORBIT 6 SEQUENCE
7	7B8	GARRETT MODE
8	847	STANDBY
		INITIALIZE MET
	C03	DACS POWER ON
12		TPM COMMAND*
13	D12	LOW POWER ON
14	E21	WARM RESTART
15		SPARE 2

Note: There are only 16 available commands and a serial 12 bit word is used. There is some logical fault detection - the command channel number (0-15) is multiplied by 111 (hex) and the result is exclusively ORed with 0CF (hex) to give the command format shown.

Reference: G. Downing memo, IOM 3464-86-229 (PASP/EDGE #35), dated 16 Oct. 1986.

*TPM COMMAND subset - the receipt of a TPM command tells the TPI board that the next command is for the TPM and should be decoded as shown below.

- 0 Reset counters to default threshold level
- 1 Increment channel 1 threshold level
- 2 Increment channel 2 threshold level
- 3 Increment channel 3 threshold level
- 4 Increment channel 4 threshold level
- 5 Increment channel 5 threshold level
- 6 Increment channel 6 threshold level
- 7 Reset counters to default threshold level

Reference: Gil Downing memo, IOM 3464-87-144 (PASP/EDGE #53), dated 11 August 1987.

4.1.2 Internal operating commands (DACS to ASIS)

4.1.2.1 Relay switching & logic control word (8 bits)

4.1.2.1.1 Bit functions

- 4.1.2.1.1.1 0, 1, 2 ARRAY SELECT
- 4.1.2.1.1.2 3, 4 I-V, HV
- 4.1.2.1.1.3 5 HV POLARITY
- 4.1.2.1.1.4 6 EMITTER POWER ON
- 4.1.2.1.1.5 7 SPARE

4.1.2.1.2 Array select commands (bits 0, 1, 2)

- 4.1.2.1.2.1 xxxxx000 NO ARRAY
- 4.1.2.1.2.2 xxxxx001 ARRAY #1
- 4.1.2.1.2.3 xxxxx010 ARRAY #2
- 4.1.2.1.2.4 xxxxx011 ARRAY #3
- 4.1.2.1.2.5 xxxxx100 ARRAY #4
- 4.1.2.1.2.6 xxxxx101 ARRAY #5
- 4.1.2.1.2.7 xxxxx110 ARRAY #6
- 4.1.2.1.2.8 xxxxx111 NO ARRAY

4.1.2.1.3 Function select commands (bits 3, 4)

- 4.1.2.1.3.1 xxx00xxx NO FUNCTION
- 4.1.2.1.3.2 xxx01xxx I-V CURVES
- 4.1.2.1.3.3 xxx10xxx HV BIAS
- 4.1.2.1.3.4 xxx11xxx HV BIAS MODE 2, HVT2

4.1.2.1.4 Bias voltage polarity commands (bit 5)

- 4.1.2.1.4.1 xx0xxxxx POS. HV BIAS
- 4.1.2.1.4.2 xx1xxxxx NEG. HV BIAS

4.1.2.1.5 Emitter select commands (bit 6)

- 4.1.2.1.5.1 x0xxxxxx EMITTER OFF
- 4.1.2.1.5.2 x1xxxxxx EMITTER ON

4.1.2.1.6 Spare (bit 7)

Reference: M. Hanna IOM 3464-87-081, dated 19 May 1987.

4.1.2.2 IV Command word (8 bits)

4.1.2.2.1 All bits used for resistance switching

4.1.2.2.2 Only 32 points selected out of 256 possible. See Table 4.1.2.2-1 below.

Table 4.1.2.2-1 - IV Curve Load Selection Commands

Load Point	Hex Command Word				
	<u>Ar. 1&5</u>	<u>Array 2</u>	<u>Ar. 3&6</u>	<u>Array 4</u>	<u>Test</u>
1	E3	2F	FF	FF	FF
2	26	5E	A1	91	81
3	42	8A	9A	DA	6A
4	4C	BC	42	22	02
5	E4	0C	AC	6C	CC
6	84	14	34	B4	D4
7	38	44	A4	E4	24
8	18	78	04	84	04
9	A8	58	B8	78	38
10	C8	E8	58	58	98
11	88	A8	E8	18	68
12	08	C8	A8	68	28
13	70	88	C8	28	C8
14	B0	08	48	48	88
15	30	F0	08	88	08
16	D0	70	F0	F0	70
17	50	B0	70	70	B0
18	90	30	B0	B0	30
19	10	D0	30	30	D0
20	E0	50	D0	D0	50
21	60	90	50	50	90
22	A0	10	90	90	10
23	20	E0	10	10	E0
24	C0	60	E0	E0	60
25	40	A0	60	60	A0
26	80	20	A0	A0	20
27	00	C0	20	20	C0
28	00	00	C0	C0	40
29	00	00	40	40	80
30	00	00	80	80	00
31	00	00	00	00	00
32	00	00	00	00	00

Reference: BASIC Calculations from G. Downing

4.1.2.3 HV Bias voltage level command (8 bits)

4.1.2.3.1 Bits 0-4 HV power supply voltage.
See Table 4.1.2.3-1 below.

Table 4.1.2.3-1 - Bias Voltage Selection Commands

Desired Bias Voltage	Command	
	Binary	Hex
0	00000000	00
50	00000011	03
100	00000110	06
150	00001001	09
200	00001100	0C
300	00010010	12
500	00011111	1F

Notes: 1. Commands for positive and negative bias voltages are identical since the polarity is switched by a relay. 2. The voltage range from 0 to 500 is divided into 31 (1F HEX) parts so each actual requested voltage is an approximation of the desired bias voltage.

4.1.2.3.2 Bits 5-7 are spares

4.2 GSE Commands

4.2.1 GSE Internal Commands - Same as PASP internal commands.

4.2.2 GSE Operating Commands

The GSE operating commands are found in the pull-down menus which are part of the GSE software and which appear automatically when the GSE controller is turned on and goes through its booting up process.

4.2.2.1 Operations Commands Menu

4.2.2.1.1 DACS Commands Menu

<u>Command</u>	<u>Key</u>
Standby command	S
orbit1 command	1
orbit2 command	2
orbit3 command	3
orbit4 command	4
orbit5 command	5
orbit6 command	6
Garrett command	G
Dacs power on	D
Warm restart	W
Low power mode	L
Initialize met	I
Quit	Q

Notes: Standby command allows presently running sequence to complete and then system waits for next command. The system automatically goes from one orbit to

the next except it stays in the occultation mode without biasing or Langmuir probe sweeps at the end of orbit 6. Garrett command turns on the occultation element sequence.

4.2.2.1.2 GSE Test Menu

<u>Command</u>	<u>Key</u>
Normal test sequence	N
Abbreviated test sequence	A
Langmuir probe data	L
Pressure gauge data	P
Sun sensor data	S
Tpm data	T
time Marker	M
Quit	Q

Note: The Abbreviated test command is functional only when the TSS plug is inserted into the system mode receptacle. The Normal test sequence is functional only when the TST plug is in place. The other commands are not yet fully functional at present.

4.2.2.1.3 Send TPM Commands Menu

<u>Command</u>	<u>Key</u>
select tpm command 0	0-7
Send tpm command	S
Quit	Q

4.2.2.1.4 Cal Commands Menu

<u>Command</u>	<u>Key</u>
select cal channel 0	0-7
ad1 mux a data packet	alt-f1
ad1 mux b data packet	alt-f2
ad2 mux a data packet	alt-f3
ad2 mux b data packet	alt-f4
Rtd data packet	R
Quit	Q

4.2.2.1.5 Telemetry Menu

<u>Command</u>	<u>Key</u>
Telemetry source RDO(TDO)	T (toggle)
save Cap file ON(OFF)	C (toggle)
Save buf file ON(OFF)	S (toggle)
View telemetry ON(OFF)	V (toggle)
Begin telemetry OFF(WAITING) (RUNNING)	B
End telemetry	E
Quit	Q

Note: The Save buf file command is not functional at present.

4.2.2.2 Development Commands Menu (No longer active)

<u>Command</u>	<u>Key</u>
----------------	------------

	Download pasp.blk	D
	make new Flight roms	F
	make new development Roms	R
	Go to forth	G
	Quit	Q
4.2.2.3	Calibration Commands Menu (Not completed)	
	<u>Command</u>	<u>Key</u>
	Asis cal menu	A
	Dacs cal menu	D
	cal File menu	F
	Quit	Q
4.2.2.3.1	Asis Cal menu (Not completed)	
	<u>Command</u>	<u>Key</u>
	select iv array # 1	1-6
	make iV1 cal table	V
	make E-telem cal table	E
	make I-telem cal table	I
	make eLect cal table	L
	make eMitter cal table	M
	cal File menu	F
	Quit	Q
4.2.2.3.2	Dacs Cal Menu	
	<u>Command</u>	<u>Key</u>
	select cal channel 0	0-7
	cal ad1 mux a	alt-f1
	cal ad1 mux b	alt-f2
	cal ad2 mux a	alt-f3
	cal ad2 mux b	alt-f4
	cal Rtd	R
	cal Dacs pmon	D
	cal Asis pmon	A
	cal File menu	F
	Quit	Q
4.2.2.4	Setup Menu	
	<u>Command</u>	<u>Key</u>
	Telemetry menu	T
	Sun sensor simulator	S
	Real time display setup	R
	Quit	Q
4.2.2.4.1	Telemetry menu - Same as menu under Operations Menu (4.2.2.1.5)	
4.2.2.4.2	Sun Sensor Simulator Menu	
	<u>Command</u>	<u>Key</u>
	1 set ssh x0 xx degrees	1
	2 set ssh xw0 xx deg/sec	2
	3 set ssh y0 xx degrees	3
	4 set ssh yw0 xx deg/sec	4
	toggle ssh Ata ON(OFF)	A
	ssh Test mode ON(OFF)	T
	Quit	Q
4.2.2.4.3	Real Time Display Setup - no menu	
5	Test Data Requirements	

- 5.1 Array output
 - 5.1.1 Digital commands used for IV curve load
 - 5.1.2 Variable load
 - 5.1.2.1 Electronic load in GSE simulators
 - 5.1.2.2 Resistive load in ASIS (IV boards)
 - 5.1.3 Analog isolation amp. for voltage telemetry
 - 5.1.4 Analog isolation amp. for current telemetry
 - 5.1.5 Analog to digital conversion (AD boards)
- 5.2 Array temperature
 - 5.2.1 RTD temperature sensors on each array
 - 5.2.2 Analog to digital conversion (RTD board)
- 5.3 Array leakage
 - 5.3.1 Digital commands used for bias voltage
 - 5.3.2 Electrometer for 10^{-3} to 10^{-7} A range
 - 5.3.3 Analog isolation amp. for electrometer
 - 5.3.4 Voltmeter for 10^{-3} to 10^{-1} A range
 - 5.3.5 Analog isolation amp. for 10^{-3} to 10^{-1} A
 - 5.3.6 Analog isolation amp. for bias voltage
 - 5.3.7 Analog to digital conversion (AD boards)
- 5.4 Sun sensor
 - 5.4.1 X-axis digital output (SSI board)
 - 5.4.2 Y-axis digital output (SSI board)
- 5.5 Langmuir probe
 - 5.5.1 Analog to dig. conv. for sweep voltage (AD boards)
 - 5.5.2 Analog to dig. conv. for electron density (AD boards)
 - 5.5.3 Analog to dig. conv. for electron temperature (AD boards)
- 5.6 Pressure Gauge
 - 5.6.1 Analog to dig. conv. for pressure (AD boards)
 - 5.6.2 Analog to dig. conv. for temp. (AD boards)
- 5.7 Transient Pulse Monitor
 - 5.7.1 Command decoding for TPM operation
 - 5.7.2 Two packets of digital data
 - 5.7.2.1 Sequence count
 - 5.7.2.2 Pulse characterization
 - 5.7.2.3 Pulse event register
 - 5.7.2.4 Additional data
 - 5.7.2.5 Housekeeping data
- 5.8 PASP housekeeping data
 - 5.8.1 Temperature (analog outputs)
 - 5.8.1.1 Power supply heat sinks (TM1,3)
 - 5.8.1.2 Housekeeping boards (TM2,4)
 - 5.8.2 Analog to dig. conv. for temp. (AD boards)
 - 5.8.3 Power supply voltage range (digital outputs of window circuits)
 - 5.8.3.1 Power supply for ASIS (HK1 board)
 - 5.8.3.1.1 +5 volts for electronics
 - 5.8.3.1.2 +5 volts for emitter
 - 5.8.3.1.3 +15 volts for isolation amps.
 - 5.8.3.1.4 +/-15 volts for electronics

- 5.8.3.1.5 +/-15 volts for electrometer
- 5.8.3.1.6 +24 volts for relays
- 5.8.3.2 Power supply for DACS (HK2 board)
 - 5.8.3.2.1 +5 volts for MOD electronics
 - 5.8.3.2.2 +5 volts for AK electronics
 - 5.8.3.2.3 +5 volts for low voltage ops.
 - 5.8.3.2.4 +5 volts for normal ops.
 - 5.8.3.2.5 +/-15 volts for normal ops.
 - 5.8.3.2.6 +28 volts (switched) from spacecraft

5.9 PASP Output Data Format

5.9.1 Source Packet Format

Each packet of source data contains 4096 bits or 512 bytes of information which consists of overhead data such as primary and secondary headers and error control and ancillary and sensor data from the PASP Plus experiment - see table 5.9-1 below.

Table 5.9-1 Telemetry Packet Format

Primary Header 48 bits			Secondary Header 80 bits			Source Data (See section 5.9.2 below)		Packet Error Control
Pack. ID	Pack. Seq.	Pack. Len.	Data Path	Time Code	Status Word #	Ancil. Data	Sensor Data	
16	16	16	16	48	16	64	3888	16

5.9.1.1 Primary Header

5.9.1.1.1 Packet ID ("0E 8E" HEX)

5.9.1.1.1.1 Version (3 bits)

5.9.1.1.1.2 Reserved (1 bit) "0"

5.9.1.1.1.3 Secondary header flag (1 bit) "1"

5.9.1.1.1.4 Application process ID (11 bits) - PASP ID is 11010001110

5.9.1.1.2 Packet Sequence Control (variable HEX)

5.9.1.1.2.1 Segmentation flags (2 bits) "11"

5.9.1.1.2.2 Source sequence count (14 bits) - modulo 16384

5.9.1.1.3 Packet Length - PASP packet length code is 253 or "00 FD" HEX

5.9.1.2 Secondary Header

5.9.1.2.1 Data Path (16 bits) ("AA AA" HEX)

5.9.1.2.1.1 Packet destination (0-7)

- 5.9.1.2.1.1.1 Real time data - "AA" HEX
- 5.9.1.2.1.1.2 Stored data - "55" HEX
- 5.9.1.2.1.2 Packet source (8-15)
- 5.9.1.2.2 Time Code (See table 5.9.1.2-1 below)

Table 5.9.1.2-1 - MET Time Code

1	6 7	12 13	19 20	26 27	36 37	46 47 48
Day	Hour	Minute	Second	mSec	1/1.024 uSec or 0-1023 Opt.	Spare
0-39	0-23	0-59	0-59	0-999		"00"

- 5.9.1.2.2.1 Day, hour, minute and second are BCD, MSB first
- 5.9.1.2.2.2 Millisecond and 1/1.024 uSec are binary coded, MSB first
- 5.9.1.2.3 Status Word Count ("6A 04" HEX)
 - 5.9.1.2.3.1 SPAS DHS information (8 bits) - "6A" HEX
 - 5.9.1.2.3.2 Status words (4 bits) For PASP "04" is used
- 5.9.2 Source Data
 - 5.9.2.1 Sun Sensor Data
 - 5.9.2.1.1 X-Axis (8 bits)
 - 5.9.2.1.2 Y-Axis (8 bits)
 - 5.9.2.1.3 ATA Settings (8 bits)
 - 5.9.2.1.4 Spare (8 bits)
 - 5.9.2.2 Spare words (32 bits)
 - 5.9.2.3 Sensor Data
 - 5.9.2.3.1 Data pointer (32 bits)
 - 5.9.2.3.2 Instrument data (3856 bits). Instrument data varies from packet to packet. In order to interpret data look for data markers and event markers as described in the data dictionary in Appendix Z.
- 5.9.3 Packet Error Control (16 bits). Packet error control is achieved by use of a two byte cyclic redundancy counter (CRC) word at the end of the packet.

- 6 Test Environmental Requirements
 - 6.1 Array current outputs (use reference solar cell)
 - 6.1.1 Collimated sunlight (calib. capability)
 - 6.1.2 Floodlights (functional test cap. only)
 - 6.2 Array voltage outputs - use same as 6.1 above
 - 6.3 Array temperatures

- 6.3.1 Use temperature chamber for calibration
- 6.3.2 Use same as 6.1.2 above for functional capability
- 6.4 Array leakage - use load resistors to 500 Megohms
- 6.5 Sun sensor X-axis
 - 6.5.1 Light input
 - 6.5.1.1 Collimated sunlight (calib. capability)
 - 6.5.1.2 Spotlight (functional test cap. only)
 - 6.5.2 Dividing head for angle measurement
- 6.6 Sun sensor Y-axis - use same as 6.5 above
- 6.7 Langmuir probe (plasma electron density & temperature)
 - 6.7.1 No environmental capability easily available
 - 6.7.2 Use furnished head simulator or GSE simulator
- 6.8 Pressure gauge
 - 6.8.1 Vacuum from 10^{-3} to 10^{-9} torr
 - 6.8.2 Temperature from -25 to +75 °C
- 6.9 Transient pulse monitor
 - 6.9.1 Arcing event
 - 6.9.1.1 GSE arc tubes might be suitable
 - 6.9.1.2 Switch induced short
 - 6.9.2 Box temperature from -25 to +75 °C

APPENDIX B - ACRONYMS

1. ASIS - Array Selection and Instrumentation System

HK1	Engineering Housekeeping Data Board #1 (ASIS)
HVC	High Voltage Power Supply Subsystem, Control Board
HVD	High Voltage Power Supply Subsystem, Driver Board
HVE	High Voltage Power Supply Subsystem, Electrometer Board
HVO	High Voltage Power Supply Subsystem, Oscillator Board
IV1	Current-Voltage Curve Subsystem, Array #1 Board
IV2	Current-Voltage Curve Subsystem, Array #2 Board
IV3	Current-Voltage Curve Subsystem, Array #3 Board
IV4	Current-Voltage Curve Subsystem, Array #4 Board
IV5	Current-Voltage Curve Subsystem, Array #5 Board
IV6	Current-Voltage Curve Subsystem, Array #6 Board
RLL	Relay Switching & Logic Subsystem, Logic Board
RLR	Relay Switching & Logic Subsystem, Relay Board
APS	Auxiliary Power Supply Board (ASIS)
PS1	Pressure Sensor #1 (ASIS)
TM1	Temperature Sensor #1 (ASIS)

2. DACS - Data Acquisition, Processing and Instrument Control System

AD1	Analog To Digital Converter Board #1
AD2	Analog To Digital Converter Board #2
CMD	Low Power Command Decoder Interface Board
DBP	DACS Back Plane Board
DCC	DACS Cardcage
DIG	Digital Input/Output Interface Board
HK2	Engineering Housekeeping Data Board #2 (DACS)
MET	Mission Elapsed Time Interface Board
RDO	Real Time Telemetry Data Output Interface Board
RTD	Resistance Thermal Detector Array Temperature Board
SEQ	Microprocessor Sequencer Control Board
SSI	Sun Sensor Data Output Interface Board
TDO	Magnetic Tape Storage Data Output Interface Board
TPI	Transient Pulse Monitor Interface Board
TER	Termination Board
SPS	Switched Power Supply Board (DACS)
PS2	Pressure Sensor #2 (DACS)
TM2	Temperature Sensor #2 (DACS)

3. GSE - Ground Support Equipment

AMCU	ASIS Manual Control Unit
AR1	Array Simulator Board for array #1 (SP)
AR2	Array Simulator Board for array #2 (GP)
AR3	Array Simulator Board for array #3 (CC)
AR4	Array Simulator Board for array #4 (SC)
AR5	Array Simulator Board for array #5 (TBD)
AR6	Array Simulator Board for array #6 (TBD)

CAL	Calibration Electronics Chassis
DVM	Digital Multimeter Chassis
DMM	Digital Multimeter
SIM	Simulator Electronics Chassis
SSE SIM	Sun Sensor Electronics Simulator Board
LPE SIM	Langmuir Probe Electronics Simulator Board
TPE SIM	Transient Pulse Monitor Electronics Simulator Board
PGE SIM	Pressure Gauge Electronics Simulator Board

4. Peripherals

LP	Langmuir Probe (AFGL)
LPE	Langmuir Probe Electronics (AFGL)
LPI	Langmuir Probe Isolation Interface Board
TPM	Transient Pulse Monitor (SRI)
TPE	Transient Pulse Monitor Electronics (SRI)
TPM S1	TPM Sensor #1 (Radiative, SRI)
TPM S2	TPM Sensor #2 (Conductive, SRI)
PG	Cold Cathode Pressure Gauge (AFGL)
PGE	Cold Cathode Pressure Gauge Electronics (AFGL)
EM	Emitter
SS	Sun Sensor
SSE	Sun Sensor Electronics
SP	Silicon Planar Array (#1)
GP	Gallium Arsenide Planar Array (#2)
CC	Mini-Cassegrainian Concentrator Array (#3)
SC	SLATS Concentrator Array (#4)

5. Miscellaneous Acronyms

A/D	Analog to digital
AFGL	Air Force Geophysics Laboratory
AIAA	American Institute of Aeronautics and Astronautics
ATA	Automatic threshold adjust
Caltech	California Institute of Technology
CAS	Contract administration service
CDR	Critical design review
cm	Centimeter
CSR	Consent to ship review
DC	Direct current
ECR	Engineering change request
ESD	Electrostatic discharge
FORTH	A dictionary-based computer language
FY	Fiscal year
GaAs	Gallium arsenide
GFE	Government Furnished Equipment
GL	Geophysics Laboratory (Air Force)
HV	High voltage
HVPS	High voltage power supply
IV	Current-voltage
I/O	Input/output
IBM	International Business Machine

ICD	Interface Control Document
IMPS	Interactions Measurement Payload for Shuttle
JPL	Jet Propulsion Laboratory
K	Kilo
kV	Kilovolts
mA	Milliamps
MIPR	Military Interdepartmental Purchase Request
MPU	Microprocessor
NASA	National Aeronautics and Space Administration
PAR	Policies and Requirements
PASP	Photovoltaic Array Space Power (Experiment)
PASP Plus	PASP Plus Diagnostics (Experiment)
PC	Personal computer
PDR	Preliminary design review
PFR	Problem/Failure Report
PIX	Plasma Interaction Experiment
QA&R	Quality Assurance and Reliability
QAP	Quality Assurance Plan
RAM	Random access memory
RS-232	A physical layer communications protocol
RSR	Resource Summary Report
RTD	Resistance Temperature Detector
SRI	SRI International, Menlo Park, California
SRM	System for Resource Management
TBD	To Be Determined
uA	Microamps
V	Volts
VDC	Volts direct current
VOLT-A	Voltage Operating Limit Tests (Experiment like PIX)
WBS	Work Breakdown Structure
WRDC	Wright Research and Development Center (Air Force)

APPENDIX C - PASP Plus INDENTED PARTS LIST

I. Mechanical Drawings

A. PASP/IMPS Drawings

- 10114662 PASP System Diagram
- 10114663 Carrier Interface - Electronic Boxes
 - 10114721 Layout, Electronic Box Assembly
 - 10114665 Final Assy - Electronic Box 1 (PEB1)
 - 10114680 Electronic Box Assy - Machined
 - 10114681 Electronic Box Fabrication
 - 10114684 Screw, Hex Head - Aluminum
 - 10114682 Cover, Electronic Box
 - 10114670 Final Assy - Card Cage
- 10114666 Final Assy - Electronic Box 2 (PEB2)
 - 10114683 Electronic Box Assy - Machined
 - 10114681 Electronic Box Fabrication
 - 10114684 Screw, Hex Head - Aluminum
 - 10125232 Backplane (see TPI sch.) [NR]
 - 10114682 Cover, Electronic Box
 - 10114671 Final Assy - Relay Box
 - 10114672 Final Assy - HiVolt Box
- 10114664 Carrier Interface - Array Plates
 - 10114667 Final Assy - Planar Array Plate (PAP)
 - 10114675 Planar Array Assembly
 - 10114673 Plate, Planar Array
 - 10114677 Bracket, Connector
 - 10114678 Langmuir Probe Mounting (LP) [NR]
 - 10114724 Langmuir Probe Boom Assembly
 - 10114720 Spacer
 - 10114722 Base Plate
 - 10114723 Probe Boom
 - Silicon Planar Array (SP)
 - Gallium Arsenide Planar Array (GP)
 - Planar Array (TBD)
 - 10114668 Final Assy - Conc. Array Plate (CAP)
 - 10114674 Concentrator Array Assembly
 - 10114676 Plate, Concentrator Array
 - 10114679 Bracket, Connector
 - Sun Sensor Head (SS)
 - Mini-Cassegrainian Con. Array (CC)
 - SLATS Concentrator Array (SC)
 - Concentrator Array (TBD)
 - TPM Antenna Sensor (TPM S1)
- 10114669 Emitter Boom (PEM)

B. PASP Plus Drawings

- 10114685 Relay Box Assy
 - 10114710 Card Cage Strap
 - 10114711 Card Cage Mount
- 10114689 Tray Assembly
 - 10114686 PCB Support Tray

- 10114687 Connector Plate
- 10114688 Insulator
- 10114713 Insulator, Thermally Conducting
- 10114725 Device Insulators
- * ASIS Printed Circuit Boards (See II.G. below)
- 10114690 Card Cage Assy
 - 10114693 Cold Plate
 - 10114695 Side Plate
 - 10114696 Board Bracket
 - 10114697 Insulator
 - 10114698 Holddown
 - 10114699 Carrier Assy
 - 10114691 Card Carrier
 - 10114692 Connector Plate
 - 10114694 Fiberglas Backing Plate
 - 10114712 Insulator, Thermally Conducting
- * DACS Printed Circuit Boards (See II.F. below)

C. Spare Numbers

- 10114700-09
- 10114714-19
- 10114645-46
- 10114756-62

II. Electronic Drawings

A. Block Diagrams

- 10125207 PASP Plus/GSE System - Simulator Op. Mode
- 10125208 PASP Plus/GSE System - System Calib. Mode
- 10125209 PASP Plus/GSE System - System Demo. Mode
- 10125214 Test Plugs

B. GSE Schematics

- 10125216 Langmuir Probe Simulator (3 pgs)
- 10125217 Sun Sensor Simulator (2 pgs)
- 10125218 ASIS Manual Control Unit
- 10125237 ARN - Array Simulator Board
- 10125261 Calibration Chassis (2 pgs)
- 10125262 Simulator Chassis (3 pgs)
- 10125263 ASIS Chassis (2 pgs)
- 10125264 DACS Chassis
- 10125265 TPM/PG Simulator Chassis
- 10125266 ASIS/DACS Pressure Sensor Simulator
- 10125267 DACS Chassis Communication Board (3 pgs)
- 10125268 DACS Chassis IBM Cont. Interface Board (4 pgs)

C. DACS Schematics

- 10125219 SEQ - 80C86 Sequence Control Board (3 pgs)
- 10125220 DIG - Digital I/O Board (2 pgs)
- 10125221 AD1 - Analog to Digital Interface Board #1 (3 pgs)
- 10125222 AD2 - Analog to Digital Interface Board #2 (3 pgs)
- 10125223 RTD - RTD Interface Board (2 pgs)

10125224 CMD - Command Decoder (3 pgs)
 10125225 SSI - Sun Sensor Interface Board (3 pgs)
 10125226 RDO - Real Time Data Interface Board (3 pgs)
 10125227 TDO - Tape Recorder Interface Board (3 pgs)
 10125228 MET - Mission Elapsed Time Interface Board (3 pgs)
 10125232 TPI - Transient Pulse Monitor Inter. Board (2 pgs)
 10125233 TER - DACS Termination
 10125234 DBP - DACS Backplane (2 pgs)

D. ASIS Schematics

10125259 SPS - Switched Power Supply Board (2 pgs)
 10125260 APS - ASIS Power Supply Board
 10125269 RLL - Relay Switching and Logic, Logic Brd (2 pgs)
 10125272 RLR - Relay Switching and Logic, Relay Board
 10125275 IVN - IV Instrumentation Board
 10125278 HVC - High Voltage Control Board (2 pgs)
 10125281 HVO - High Voltage Oscillator Board
 10125284 HVE - High Voltage Electrometer Board
 10125299 HVD - High Voltage Driver Board

E. Miscellaneous Schematics

10125229 HK1 - Housekeeping Board #1 (2 pgs)
 10125230 HK2 - Housekeeping Board #2 (2 pgs)
 10125231 LPI - Langmuir Probe Isolation Board

F. DACS Printed Circuit Boards

10125240 SEQ - Microprocessor Sequence Control Brd (8 pgs)
 10125241 DIG - Digital Input/Output Board (7 pgs)
 10125242 AD1 - Analog to Digital Converter #1 (9 pgs)
 10125243 RTD - Resistance Thermal Detector Array (8 pgs)
 10125244 CMD - Low Power Command Decoder (8 pgs)
 10125245 SSI - Sun Sensor Data Output (9 pgs)
 10125246 RDO - Real Time Data Output (8 pgs)
 10125247 TDO - Tape Storage Data Output (8 pgs)
 10125248 MET - Mission Elapsed Time (8 pgs)
 10125251 DBP - DACS Back Plane Board (7 pgs)
 10125252 TER - Termination Board (8 pgs)
 10125255 TPI - Transient Pulse Monitor (8 pgs)
 10125274 AD2 - Analog to Digital Converter #2 (5 pgs)

G. ASIS Printed Circuit Boards

10125249 HK - Housekeeping Board (10 pgs)
 10125257 SPS - Switched Power Supply (8 pgs)
 10125258 APS - Auxiliary Power Supply (7 pgs)
 10125270 RLL - Relay Switching and Logic, Logic Brd (8 pgs)
 10125271 HK2 - Housekeeping Board #2 (6 pgs)
 10125273 RLR - Relay Switching and Logic, Relay Brd (7 pgs)
 10125276 IV - I vs. V Measurement (7 pgs)
 10125279 HVC - High Voltage Power Supply Cont. Brd (9 pgs)
 10125282 HVO - Hi Voltage Pow. Supp. Oscillator Brd (8 pgs)
 10125291 High Voltage Transformer
 10125289 Assembly, Terminal Board

10125285 HVE - Hi Volt. Pow. Supp. Electrometer Brd (8 pgs)
 10125294 IV2 - I vs. V Measurement (3 pgs)
 10125295 IV3 - I vs. V Measurement (3 pgs)
 10125296 IV4 - I vs. V Measurement (3 pgs)
 10125297 IV5 - I vs. V Measurement (3 pgs)
 10125298 IV6 - I vs. V Measurement (3 pgs)
 10125300 HVD - High Volt. Power Supply Driver Board (7 pgs)
 10125290 Drive transformer
 10125288 Assembly, Terminal Board
 10125292 Output Choke
 10125293 Input Choke
 H. Miscellaneous Printed Circuit Boards
 10125254 LPI - Langmuir Probe Interface (7 pgs)
 10125250 Langmuir Probe Board Artwork (not drawn)
 10125256 AR - Array Simulator Board (8 pgs)

I. Wire Net Lists

10125303 RELWIR Cabling Wire Net List (2 pgs)
 10136753 APS Circuit Board Wire Net List (2 pgs)
 10136754 ARN Circuit Board Wire Net List
 10136755 HKN Circuit Board Wire Net List (5 pgs)
 10136756 HVC Circuit Board Wire Net List (4 pgs)
 10136757 HVD Circuit Board Wire Net List (2 pgs)
 10136758 HVE Circuit Board Wire Net List (2 pgs)
 10136759 HVO Circuit Board Wire Net List (2 pgs)
 10136760 IVN Circuit Board Wire Net List (2 pgs)
 10136761 RLL Circuit Board Wire Net List (4 pgs)
 10136762 RLR Circuit Board Wire Net List (3 pgs)
 10136763 SPS Circuit Board Wire Net List (4 pgs)
 10136764 ADN Circuit Board Wire Net List (5 pgs)
 10136765 LPI Circuit Board Wire Net List (2 pgs)
 10136766 CMD Circuit Board Wire Net List (4 pgs)
 10136767 DBP Circuit Board Wire Net List (7 pgs)
 10136768 DIG Circuit Board Wire Net List (3 pgs)
 10136769 MET Circuit Board Wire Net List (4 pgs)
 10136770 RDO Circuit Board Wire Net List (5 pgs)
 10136771 RTD Circuit Board Wire Net List (4 pgs)
 10136772 SEQ Circuit Board Wire Net List (5 pgs)
 10136773 SSI Circuit Board Wire Net List (5 pgs)
 10136774 TDO Circuit Board Wire Net List (4 pgs)
 10136775 TER Circuit Board Wire Net List (3 pgs)
 10136776 TPI Circuit Board Wire Net List (4 pgs)

J. Cabling Drawings

10125301 GSE Cables, ASIS Cabling Harness (not drawn)
 10125302 GSE Cables, DACS Cabling Harness (not drawn)
 10136777 GSE Cables, ASIS/DACS Analog Cable
 10136778 GSE Cables, ASIS/DACS Digital Cable
 10136779 GSE Cables, Simulator/ASIS Array Cable
 10136780 GSE Cables, Simulator/DACS RTD Cable
 10136781 GSE Cables, ASIS/Simulator Emitter Cable

10136782 GSE Cables, Simulator/DACS Langmuir Probe Cable
10136783 GSE Cables, Simulator/DACS Sun Sensor Cable
10136784 GSE Cables, IMPS Sun Sensor Cable
10136785 GSE Cables, SIM/DACS 28V Power Cable
10136786 GSE Cables, IMPS Signal Cable
10136787 GSE Cables, ASIS 28V Switched Power Cable
10136788 GSE Cables, DACS Power Cable
10136789 ASIS Cables, Analog Cable
10136790 ASIS Cables, IV-RLR Cable
10136791 ASIS Cables, Emitter & IV-RLR Cable
10136792 ASIS Cables, DIG Cable
10136793 ASIS Cables, HK1 Power Monitor Cable
10136794 DACS Cables, RTD Cable
10136795 DACS Cables, IMPS Cable
10136796 DACS Cables, TPM Cables
10136797 DACS Cables, DBP Cable
10136798 DACS Cables, HK2 Power Monitor Cable
10136799 DACS Cables, Analog Cable
10136800 DACS Cables, DIG Cable
10136801 DACS Cables, SSH Cable
10136802 DACS Cables, Sun Sensor Cable

K. Spare Numbers

10125204-06
10125210-13
10125215
10125235-36
10125238-39
10125253
10125264
10125277
10125280
10125283
10125286-87

APPENDIX D - PASP Plus SYSTEM TEST PROCEDURES

1 Introduction

These system test procedures for the PASP Plus Diagnostics experiment define the test protocols for performing a functional demonstration level of testing. The controlling document is the "PASP Plus Diagnostics Test Objectives". This test procedure document contains a response to every paragraph of the test objectives document.

2 Subsystem Level Test Procedures

2.1 Solar Arrays

2.1.1 Electrical

2.1.1.1 Array Operation Was not requested.

2.1.1.2 Array Isolation Previously done in array acceptance testing.

2.1.2 Thermal Was not requested.

2.1.3 Mechanical (concentrator arrays only) Was not requested.

2.1.4 Optical (concentrator arrays only) Was not requested.

2.1.5 Life Test Was not requested since PASP Plus test objectives only require two to four weeks of operation.

2.2 Sun Sensor

2.2.1 Electrical Has already been done for Sun Sensor Acceptance Test.

2.2.2 Mechanical Has already been done for Sun Sensor Acceptance Test.

2.3 Cold Cathode Pressure Gauge (PG) Was not requested.

2.4 Transient Pulse Monitor (TPM) Was not requested.

2.5 Langmuir Probe (LP)

2.5.1 Sweep Voltage

2.5.1.1 Connect LP head simulator to LP electronics (LPE). Note: Langmuir probe head test will not be performed.

2.5.1.2 Connect LPE output to oscilloscope.

2.5.1.3 Turn on LPE and the LP head simulator and measure sweep voltage wave shape and voltage levels.

- 2.5.1.4 Compare sweep wave shape and voltage levels with calibrated data.
- 2.5.2 Electron Density (N_e)
Was not requested.
- 2.5.3 Electron Temperature (T)
Was not requested.
- 2.6 Emitter
Was not requested.
- 2.7 GSE Subsystem
 - 2.7.1 Verify operation and range of all power supplies.
 - 2.7.2 Verify proper voltage on all wafer switches at all switch positions.
 - 2.7.3 Verify functionality of all controls, displays and meters.
- 2.8 PASP Plus Subsystem
 - 2.8.1 Array Switching
 - 2.8.1.1 Array Selection
 - 2.8.1.1.1 Connect PASP Plus subsystem to GSE.
 - 2.8.1.1.2 Use GSE to command selection of Array #1 and record if Array #1 GSE simulator indicates proper selection.
 - 2.8.1.1.3 Repeat above for Arrays #2 through #6.
 - 2.8.1.2 Array Frame Grounding (concentrator arrays only)
 - 2.8.1.2.1 With PASP Plus connected to the GSE, select the HVT2 mode and command the frame of Array #3 to be grounded to the array negative power lead.
 - 2.8.1.2.2 Record if array frame is properly grounded.
 - 2.8.1.2.3 Repeat steps 2.8.1.2.1 and 2.8.1.2.2 above for arrays 4 and 6.
 - 2.8.2 IV Curve
 - 2.8.2.1 Current Axis
 - 2.8.2.1.1 Set the METER FUNCTION selector switch on the CAL chassis of the GSE to IV-I.
 - 2.8.2.1.2 Set the ARRAY/RTD/HV LOAD selector switch to Array #1. When the 0-1 amp current power supply control is set to a particular value, that value will be displayed on DMM #1. The corresponding ASIS analog telemetry output voltage will be displayed on DMM #2.
 - 2.8.2.1.3 Initiate the measurement process by calling the calibration menu and selecting the appropriate parameters.
 - 2.8.2.1.4 Manually sweep the power supply control through the range of interest as the GSE controller collects the data.
 - 2.8.2.1.5 Repeat steps 2.8.2.1.3 and 2.8.2.1.4 above with the ARRAY/RTD/HV LOAD selector switch set to the Array #2 through Array #6 positions.

2.8.2.1.6 Compare the applied current values with the corresponding analog telemetry values.

2.8.2.2 Voltage Axis

2.8.2.2.1 Set the METER FUNCTION selector switch to IV-V.

2.8.2.2.2 Set the ARRAY/RTD/HV LOAD selector switch to Array #1. When the 20-120 volt power supply control is set to a particular value, that value will be displayed on DMM #1. The corresponding ASIS analog telemetry output voltage will be displayed on DMM #2.

2.8.2.2.3 Initiate the measurement process by calling the calibration menu and selecting the appropriate parameters.

2.8.2.2.4 Manually sweep the power supply control through the range of interest as the controller collects the data.

2.8.2.2.5 Repeat the above process with the ARRAY/RTD/HV LOAD selector switch set to the Array #2 position.

2.8.2.2.6 Repeat steps 2.8.2.2.2. through 2.8.2.2.4. for Array #3 through Array #6 using the 0-20 volt power supply in place of the 20-120 volt power supply.

2.8.2.2.7 Compare the applied voltage values with the corresponding analog telemetry values.

2.8.3 High Voltage Power Supply

2.8.3.1 Voltage Level (ETELEM channel)

2.8.3.1.1 Set the METER FUNCTION selector switch to HV-V.

2.8.3.1.2 Set the ARRAY/RTD/HV LOAD selector switch to Array #1 position which will apply a 50K ohm load across the high voltage supply.

2.8.3.1.3 Initiate the measurement process by calling the calibration menu and selecting the appropriate parameters.

2.8.3.1.4 The controller will step the ASIS high voltage power supply and collect the data of the actual output voltage (DMM 1) and the corresponding ASIS analog telemetry output voltage (DMM 2) across the appropriate high voltage load.

2.8.3.1.5 Repeat 2.8.3.1.1 through 2.8.3.1.4 for the ARRAY/RTD/HV LOAD selector switch in the Array #2 through Array #6 positions. This will apply a 25K, 10K, 5K, and 4.5K ohm load respectively across the high voltage supply. Compare the output voltage values with the corresponding analog telemetry values. The output voltage is normally affected slightly with changing load. However, when the Array #5 position is selected the high voltage supply over-current protection circuitry should disable the high voltage supply for two seconds when it is stepped to 500 V.

2.8.3.2 Noise

2.8.3.2.1 Attach an oscilloscope to the output from the high voltage bias supply.

- 2.8.3.2.2 Select the first positive bias voltage step value and measure the corresponding peak-to-peak noise.
- 2.8.3.2.3 Repeat above for remaining selected positive and negative bias voltage values.
- 2.8.3.2.4 Compare measured noise with noise specifications.
- 2.8.3.3 Polarity
Use oscilloscope to verify polarity of applied bias voltage of high voltage supply.
- 2.8.3.4 Leakage Current (ITELEM channel, high leakage current range)
 - 2.8.3.4.1 Set the METER FUNCTION selector switch to HV-I.
 - 2.8.3.4.2 Set the ARRAY/RTD/HV LOAD selector switch to Array #1 which will apply a 50K ohm load across the high voltage supply.
 - 2.8.3.4.3 Initiate the measurement process by calling the calibration menu and selecting the appropriate parameters.
 - 2.8.3.4.4 The controller will step the ASIS high voltage power supply and collect the data of actual leakage current (DMM #1) and corresponding analog telemetry output voltage (DMM #2) across the appropriate high voltage load.
 - 2.8.3.4.5 Repeat steps 2.8.3.4.1 through 2.8.3.4.4 for the ARRAY/RTD/HV LOAD selector switch in the Array #2 through Array #5 positions. This will apply a 25K, 10K, 5K, and 4.5K ohm load respectively across the high voltage supply.
 - 2.8.3.4.6 Compare the actual leakage current output values with the corresponding analog telemetry values. The output current will vary with applied voltage and load resistance. However, when the Array #5 position is selected the high voltage supply over-current protection circuitry should disable the high voltage supply for two seconds when it is stepped to 500 V.
- 2.8.4 Electrometer (ELECT channel, low leakage current range)
 - 2.8.4.1 Set the METER FUNCTION selector switch to ELECT.
 - 2.8.4.2 Set the ARRAY/RTD/HV LOAD selector switch to Array 6/Elect. The input current determined by the setting of the ELECTROMETER CURRENT selector switch will be indicated on DMM #1. The corresponding ASIS analog telemetry output voltage will be indicated on DMM #2.
 - 2.8.4.3 Initiate the measurement process by calling the calibration menu and selecting the appropriate parameters.
 - 2.8.4.4 Step the ELECTROMETER CURRENT selector switch through the full range of values as the controller collects the data.
 - 2.8.4.5 Compare the applied current values with the corresponding analog telemetry values.

- 2.8.5 Emitter (EMITTER channel)
 - 2.8.5.1 Set the METER FUNCTION selector switch to EMITTER.
 - 2.8.5.2 Set the ARRAY/RTD/HV LOAD selector switch to Off. The emission current determined by the 0-100 mA EMISSION power supply will be indicated on DMM #1. The corresponding ASIS analog telemetry voltage output will be indicated on DMM #2.
 - 2.8.5.3 Initiate the measurement process by calling the calibration menu and selecting the appropriate parameters.
 - 2.8.5.4 Sweep the EMITTER emission current power supply through the full range as the controller collects the data.
 - 2.8.5.5 Compare the applied current values with the corresponding analog telemetry values.
- 2.8.6 RTD Temperature Sensors
 - 2.8.6.1 Set the ARRAY/RTD/HV LOAD selector switch to RTD #1.
 - 2.8.6.2 Set the RTD SIMULATOR selector switch to the desired resistance value.
 - 2.8.6.3 Initiate the measurement process by calling the calibration menu and selecting the appropriate parameters.
 - 2.8.6.4 Repeat the above process with the ARRAY/RTD/HV LOAD selector switch set to the RTD #2 through RTD #6 positions.
 - 2.8.6.5 Compare the selected resistance values with the corresponding digital telemetry values and the appropriate RTD calibration tables.
- 2.8.7 Analog/Digital Conversion
 - 2.8.7.1 Set the METER FUNCTION selector switch to AD.
 - 2.8.7.2 Initiate the measurement process by calling the calibration menu and selecting the appropriate parameters.
 - 2.8.7.3 Manually sweep the -5v/+5v power supply through the full range as the controller collects the data. The actual applied voltage will be indicated on DMM #1 and the analog telemetry voltage will be shown on DMM #2.
 - 2.8.7.4 Perform 2.8.7.2 and 2.8.7.3 for all of the 32 A/D channels.
 - 2.8.7.5 Compare the actual applied voltage values with the corresponding digital telemetry values.
- 2.8.8 Sun Sensor Interface
 - 2.8.8.1 Connect the sun sensor simulator in the GSE to the DACS.
 - 2.8.8.2 Send 0, 0, OFF data from sun sensor simulator to DACS.
 - 2.8.8.3 Repeat above with 80, 80, ON data from simulator.
 - 2.8.8.4 Compare simulated data to digital telemetry data.
- 2.8.9 Pressure Sensor
 - Was not requested.

- 2.8.10 Housekeeping Data
- 2.8.10.1 Power Supply Voltages
- 2.8.10.1.1 Remove voltage sensing plug from the housekeeping circuit board in the ASIS electronics (HK1) and replace with a test plug.
- 2.8.10.1.2 Apply low and high test voltages to the window circuit of channel 0 on the HK1 board while collecting housekeeping data. See Table 2.8.10.1-1 below for appropriate test voltages.
- 2.8.10.1.3 Repeat above for channels 1 through 7 of HK2.

Table 2.8.10.1.-1 - Housekeeping Voltage Ranges

HK1, Connector J-2

Chan Num.	Pin +	Num. -	Power Supply	Voltage Range		
				Nom.	High	Low
0	2	21	+5V EMIT	5.0	5.5	4.5
1	4	23	+5V ASIS	5.0	5.5	4.5
2	6	25	+15V ASIS	15.0	16.5	13.5
3	27	8	-15V ASIS	15.0	16.5	13.5
4	10	29	+24V RLY	24.0	26.4	21.6
5	12	31	+15VAD210	15.0	16.5	13.5
6	33	14	-15V HVE	15.0	16.5	13.5
7	16	35	+15V HVE	15.0	16.5	13.5

HK2, Connector J-2

Chan Num.	Pin +	Num. -	Power Supply	Voltage Range		
				Nom.	High	Low
0	2	21	+5V AK	5.0	5.5	4.5
1	4	23	+5V MO	5.0	5.5	4.5
2	6	25	+5V LO	5.0	5.5	4.5
3	8	27	+5V DAC	5.0	5.5	4.5
4	10	29	+15V DAC	15.0	16.5	13.5
5	31	12	-15V DAC	15.0	16.5	13.5
6	14	33	+28V SW	28.0	30.8	25.2
7	16	35	+5V SPARE	5.0	5.5	4.5

- 2.8.10.1.4 Compare applied low and high window voltages with desired window voltages as shown in Table 2 above.
- 2.8.10.1.5 Repeat 2.8.10.1.1 through 2.8.10.1.4 for the HK2 board located in the DACS.
- 2.8.10.2 Housekeeping Temperatures
Was not requested.
- 2.8.11 DACS Commands
- 2.8.11.1 Response
- 2.8.11.1.1 Run through entire set of DACS system commands by manually keying in commands.
- 2.8.11.1.2 Compare actual PASP Plus system response to commands with expected response.

- 2.8.11.2 Sequence
Use data generated in step 2.8.11.1.1 above and compare actual event sequence and timing with expected events and timing.
- 2.8.12 Transient Pulse Monitor (TPM)
 - 2.8.12.1 Commands
Use data obtained in step 2.8.11.1.1 above to see if all TPM commands were properly received and decoded by the transient pulse monitor interface board (TPI).
 - 2.8.12.2 Buffering
Use data from 2.8.11.1.1 above to check on buffering of TPM data and inclusion in DACS to spacecraft data stream.
- 2.8.13 Telemetry
Use data from 2.8.11.1.1 above to check formatting of DACS telemetry data versus specified data format.
- 2.8.14 Subsystem Thermal Test (Optional test)
The overall plan of this portion of the test procedure is to run a baseline test using an abbreviated, computer-controlled test sequence with simulated input signals for each sensor at room temperature (+25 °C) and then run verification tests at low and high system operating temperatures. Examining the outputs obtained from these tests will disclose the output differences made by the temperature changes. The abbreviated test sequence contains all of the operating commands and data collection of a normal flight data acquisition element sequence (see Figures 1 and 2) except that the 20 second application of each of the voltage bias steps has been reduced to less than a second. This allows a 17 minute normal flight data acquisition element sequence to be run in about 3.5 minutes.
 - 2.8.14.1 Place ASIS and DACS electronics from the GSE into a temperature chamber.
 - 2.8.14.2 Make sure that the abbreviated test software is loaded on the controller hard disk.
 - 2.8.14.3 Set up a buffer file named ATS1 (Abbreviated Test Sequence #1).
 - 2.8.14.4 Enter xxxx into controller and run an abbreviated test sequence at ambient temperature (+25 °C).
 - 2.8.14.5 Gradually lower test chamber temperature to 0 °C and repeat above except buffer file is named ATS2.
 - 2.8.14.6 Gradually raise test chamber temperature to +50 °C and repeat above except buffer file is named ATS3.
 - 2.8.14.7 Compare test results with previous data and then remove equipment from chamber.
 - 2.8.14.8 Repeat steps 2.8.14.1 through 2.8.14.7 except use brassboard ASIS and DACS electronics and use buffer files

named ATS4, ATS5, and ATS6.

3 Interface Tests

3.1 Mechanical

- 3.1.1 Check for presence and proper location of all system components.
- 3.1.2 Check for proper mounting of all system components.
Note: unless specifically required all mounting hardware is to be only finger tight during system tests so as to prevent galling or damage to flight equipment.

3.2 Connectors

3.2.1 General

- 3.2.1.1 Check connectors for presence of proper labelling before mating.
- 3.2.1.2 Visually inspect all connector pins for straightness, alignment and contamination.

3.2.2 Power Connectors

- 3.2.2.1 Before mating power connectors check voltage level and polarity and record results on a log sheet (See Table 3.2.2-1 at end of this document). Note: this table has been deleted and Appendix S is substituted.
- 3.2.2.2 Compare logged values with correct power voltage conditions and then initial and date log sheet unless values do not coincide.

3.2.3 Signal Connectors

Check continuity and pin assignment against schematic data.

4 System Tests

4.1 GSE-Based Operation Tests

The overall plan of the GSE-based operation portion of the baseline test procedure is to run a baseline test using a computer-controlled test sequence with low level or no simulated input signals for each sensor and then run a verification test with high level simulated input signals. Examining the outputs obtained from these two tests will disclose the output differences made by the simulated inputs. The computer-controlled test sequence contains all of the operating commands and data collection of a normal flight orbit sequence (see Figures 1 and 2) and runs for about 35 minutes.

4.1.1 Test Setup

- 4.1.1.1 Wire system per block diagram 10125207.
- 4.1.1.2 Make sure that the test software is loaded on the controller hard disk.
- 4.1.1.3 Place all arc lamp switches in the OFF position.
- 4.1.1.4 Place Pressure Gauge (PG) simulators for pressure and temperature in the low mode.

- 4.1.1.5 Place the TPM simulator in its initial condition.
- 4.1.1.6 Set the sun sensor head simulator to 0, 0, OFF.
- 4.1.2 GSE Baseline Test (Orbit Test Sequence #1)
 - 4.1.2.1 Set up buffer file named OTS1.
 - 4.1.2.2 Enter xxxx into controller and run test.
 - 4.1.2.3 Test Outputs
 - 4.1.2.3.1 AR1 through AR6 simulator IV curve current and voltage data and array temperatures as seen through IV1 through IV6 printed circuit board sensors and DACS AD convertor channels.
 - 4.1.2.3.2 Array bias voltages.
 - 4.1.2.3.3 Plasma leakage current for each array bias voltage.
 - 4.1.2.3.4 PG pressure and temperature data for low level input voltages.
 - 4.1.2.3.5 TPM channel number, event register count and pulse characterization data for E-field and current sensor channels.
 - 4.1.2.3.6 Sun sensor gray scale reading for OFF.
- 4.1.3 GSE Verification Test (Orbit Test Sequence #2)
 - 4.1.3.1 Test Condition Modifications
 - 4.1.3.1.1 Set +300 V and -400 V arc lamp switches to ON.
 - 4.1.3.1.2 Set PG simulators to high mode for both pressure and temperature.
 - 4.1.3.1.3 Place E-field stimulator on E-field antenna and simulate an arc event E-field.
 - 4.1.3.1.4 Set sun sensor simulator to 80, 80, ON.
 - 4.1.3.1.5 Set up buffer file named OTS2.
 - 4.1.3.1.6 Enter xxxx into controller and run test.
 - 4.1.3.2 Test Outputs
 - 4.1.3.2.1 AR1 through AR6 simulator IV curve data and higher array temperatures.
 - 4.1.3.2.2 Array bias voltages with over-current events at +300 and -400 V.
 - 4.1.3.2.3 PG pressure and temperature data for high level input voltages.
 - 4.1.3.2.4 TPM channel number, event register count and pulse characterization data .
 - 4.1.3.2.5 Sun sensor gray scale reading for ON.
- 4.1.3.3 Test Data Analysis
 - Compare data with similar data obtained in PASP Plus subsystem tests (see step 2.8).

4.2 PASP Brassboard Tests

The PASP brassboard system tests will exercise all of the functions called out in paragraph 4.1 above. The only difference is the use of the ASIS and DACS brassboard electronics mounted on the system demonstration cart in place of the built-in GSE electronics.

4.2.1 Test Setup
System wiring per block diagram 10125207. Note: the ASIS and DACS electronics along with the pressure sensors will be located on the demonstration test cart.

4.2.2 Test Conditions
Repeat paragraphs 4.1.1.2 through 4.1.3.3 above except the data buffer files are to be named OTS3 and OTS4.

4.3 System Level Equipment Tests

The system level baseline tests exercise all peripheral arrays and sensors and all channels of the PASP Plus Diagnostics electronics. However, some sensor inputs will be simulated rather than stimulated since certain environmental inputs are difficult or expensive to obtain at a system test level. These simulated inputs will be identified in the detailed test procedures below.

The overall plan of the flight equipment system tests is to run a baseline test using an abbreviated, computer-controlled test sequence with low level or no simulated input signals for each sensor. After the baseline test another test sequence will be run with sensors stimulated or simulated. Examination of the outputs from the two test sequences allows the determination of changes in the output data caused by changes in the inputs of each channel. The numerical data that is obtained will be qualitative in most cases, not quantitative. There are two reasons for this lack of quantitative data: first, the inputs from various stimulators such as floodlights are not controlled or referenced; second, the data has not been subsequently processed using a calibration table in order to obtain actual values. This is only a small loss since the actual values of the array and sensor outputs from known inputs have previously been well characterized during acceptance tests performed at JPL or at Air Force facilities or from earlier subsystem testing. This earlier testing along with the results of calibration tests should provide adequate assurance of proper operation of the electronic systems.

4.3.1 Array Tests

4.3.1.1 Array Operation

4.3.1.1.1 Wire system per block diagram 10125209.

4.3.1.1.2 Assemble PASP Plus brassboard electronics and flight equipment on system demonstration cart per Figures 3 and 4 and using array mounting torque values shown below.

Table 1 - Array Mounting Torques

SP Array	40 in-lbs.
GP Array	40 in-lbs.

CC Array 25 in-lbs.
SC Array 110 in-lbs.

- 4.3.1.1.3 Make sure that the abbreviated test software is loaded on the controller hard disk.
- 4.3.1.1.4 Remove protective covers from arrays.
- 4.3.1.1.5 Set up buffer file named ATS7.
- 4.3.1.1.6 Turn on floodlights.
- 4.3.1.1.7 Enter xxxx into the controller and run test immediately after floodlights are turned on.
- 4.3.1.1.8 Array #1 through #6 IV curve data will be obtained by controller in buffer file.
- 4.3.1.1.9 Turn off floodlights and replace array protective covers.
- 4.3.1.1.10 Move a reference solar cell over the area of each array and measure the open circuit voltage of the reference solar cell to obtain approximate light level for each array. Record light levels.
- 4.3.1.1.11 Compare test data with acceptance test or subsystem test data after making corrections for light level.
- 4.3.1.2 Array Isolation
 - 4.3.1.2.1 Data obtained in 4.3.1.1.8 contains high voltage bias (ELEM) and array (or plasma) leakage (I-ELEM & E-ELEM) data for Arrays #1 through #6.
 - 4.3.1.2.2 Compare test data with acceptance test or subsystem test data.
- 4.3.1.3 Thermal
 - 4.3.1.3.1 Rerun test per steps 4.3.1.1.5 through 4.3.1.1.8 except set up buffer file named ATS8 and wait five minutes after floodlights are turned on before running test.
 - 4.3.1.3.2 Data obtained in 4.3.1.1.8 contains initial temperature (RTD) data for Arrays #1 through #6 at a level slightly above room temperature.
 - 4.3.1.3.3 Data obtained in 4.3.1.3.1 contains temperature data for Arrays #1 through #6 at a level well above room temperature.
 - 4.3.1.3.4 Compare both sets of test data with acceptance test or subsystem test data.
 - 4.3.1.3.5 Turn off floodlights and replace array protective covers.
- 4.3.2 Sensor Tests
 - 4.3.2.1 Sun Sensor
 - 4.3.2.1.1 Place spotlight about 3 feet from sun sensor head and align as close to normal with sun sensor head as possible.
 - 4.3.2.1.2 Remove sun sensor head protective cover.
 - 4.3.2.1.3 Set up buffer file named ATS9.
 - 4.3.2.1.4 Turn on spotlight.
 - 4.3.2.1.5 Enter xxxx into controller and immediately run test.
 - 4.3.2.1.6 Move spotlight so it is aligned as near to 15° from both X and Y axes.

- 4.3.2.1.7 Set up buffer file named ATS10.
- 4.3.2.1.8 Enter xxxx into controller and run test.
- 4.3.2.1.9 Data obtained in step 4.3.2.1.5 above contains sun sensor X- and Y-axis outputs near zero degrees while data obtained from 4.3.2.1.8 contains sun sensor data near 15°.
- 4.3.2.1.10 Compare data from both tests with acceptance test or subsystem test data.
- 4.3.2.1.11 Turn off spotlight and replace sun sensor head protective cover.

- 4.3.2.2 Cold Cathode Pressure Gauge (PG)
 - 4.3.2.2.1 Disconnect PG sensor head from PGE and connect GSE PG simulator.
 - 4.3.2.2.2 Set PG simulator switches to low level for both pressure and temperature channels.
 - 4.3.2.2.3 Set up buffer file named ATS11.
 - 4.3.2.2.4 Enter xxxx into controller and run test.
 - 4.3.2.2.5 Set PG simulator switches to high level for both pressure and temperature channels.
 - 4.3.2.2.6 Set up buffer file named ATS12.
 - 4.3.2.2.7 Enter xxxx into controller and run test.
 - 4.3.2.2.8 Compare data from both tests with acceptance test or subsystem test data.
 - 4.3.2.2.9 Reconnect PG head to PGE.

- 4.3.2.3 Transient Pulse Monitor (TPM)

A current pulse sensor test was not requested. The E-field sensor test procedure is given below.

 - 4.3.2.3.1 Place E-field stimulator over E-field antenna and stimulate E-field.
 - 4.3.2.3.2 Set up buffer file named ATS13.
 - 4.3.2.3.3 Enter xxxx into controller and run test.
 - 4.3.2.3.4 Compare data from 4.3.2.3.3 with data obtained in step 4.3.2.2.6 above to see changes in event register.
 - 4.3.2.3.5 Compare pulse characterization data obtained in step 4.3.2.3.3 with acceptance test or subsystem test data.

- 4.3.2.4 Langmuir Probe (LP)
 - 4.3.2.4.1 Connect Langmuir probe head simulator in GSE to Langmuir probe electronics (LPE).
 - 4.3.2.4.2 Set up buffer file named ATS14.
 - 4.3.2.4.3 Enter xxxx into controller and run test.
 - 4.3.2.4.4 Compare data from above with acceptance test or subsystem test data.

- 4.3.2.5 Emitter
 - 4.3.2.5.1 Connect emitter simulator in GSE with ASIS electronics.
 - 4.3.2.5.2 Set up buffer file named ATS15.
 - 4.3.2.5.3 Enter xxxx into controller and run test.
 - 4.3.2.5.4 Compare data from above with subsystem test data.

- 4.3.2.6 Pressure Sensors
Was not requested.
- 4.3.3 Housekeeping Tests
 - 4.3.3.1 Power Supply Voltages
Was not requested.
 - 4.3.3.2 Temperature
Was not requested.
- 4.3.4 DACS Command Tests
 - 4.3.4.1 Response
 - 4.3.4.1.1 Run through entire set of DACS system commands by manually keying in commands.
 - 4.3.4.1.2 Compare actual PASP Plus system response to commands with expected response.
 - 4.3.4.2 Sequence
 - 4.3.4.2.1 Set up buffer file named OTS5.
 - 4.3.4.2.2 Run system through a complete six orbit data sequence from flight PROM chips.
 - 4.3.4.2.3 Compare actual PASP Plus system sequence with expected sequence.
 - 4.3.4.3 TPM Commands
Use data obtained in step 4.3.4.2.2 above to see if all TPM commands were properly received and decoded by the transient pulse monitor interface board (TPI).
 - 4.3.4.4 Buffering
Use data from 4.3.4.2.2 above to check on buffering of TPM data and inclusion in DACS to spacecraft data stream.
 - 4.3.4.5 Telemetry
Use data from 4.3.4.2.2 above to check formatting of DACS telemetry data versus specified data format for both real time (RDO) and tape recorder (TDO) output channels.
- 4.3.5 Integrated System Test
 - 4.3.5.1 Input Power
 - 4.3.5.1.1 Measure and record in-rush current when system is turned on.
 - 4.3.5.1.2 Measure and record noise on input power line when high voltage power supply is operating.
 - 4.3.5.1.3 Compare data from tests with system requirements.
 - 4.3.5.2 Input Voltage
 - 4.3.5.2.1 Make sure that the abbreviated test software is loaded on the controller hard disk.
 - 4.3.5.2.2 Make sure input voltage is at 28 Volts DC.
 - 4.3.5.2.3 Set up buffer file named ATS16.
 - 4.3.5.2.4 Enter xxxx into controller and run test.

- 4.3.5.2.5 Change input voltage to 25 VDC.
 - 4.3.5.2.6 Set up buffer file named ATS17.
 - 4.3.5.2.7 Enter xxxx into controller and run test.
 - 4.3.5.2.8 Change input voltage to 32 VDC.
 - 4.3.5.2.9 Set up buffer file named ATS18.
 - 4.3.5.2.10 Enter xxxx into controller and run test.
 - 4.3.5.2.11 Compare data from all three tests with system requirements.
- 4.3.5.3 System Thermal Test
Was not requested.

APPENDIX E - PASP Plus SYSTEM TEST REPORT

1 Introduction

This system test report for the PASP Plus Diagnostics Experiment is formatted the same as the PASP Plus Baseline Test Procedures functional demonstration test document. The controlling document is the "PASP Plus Diagnostics Test Objectives". This test report document contains a response to every paragraph of the test objectives document.

2 Subsystem Level Test Results

2.1 Solar Arrays

2.1.1 Electrical

2.1.1.1 Array Operation

Not requested since this test was done previously during array acceptance testing.

2.1.1.2 Array Isolation

Not requested since this test was done previously during array acceptance testing.

2.1.2 Thermal

Not requested.

2.1.3 Mechanical (concentrator arrays only)

Not requested.

2.1.4 Optical (concentrator arrays only)

Not requested.

2.1.5 Life Test

Not requested since PASP Plus test objectives are based upon only two to four weeks of operation.

2.2 Sun Sensor

2.2.1 Electrical

Not requested since this test was done previously during the Sun Sensor Acceptance Test.

2.2.2 Mechanical

Not requested since this test was done previously during the Sun Sensor Acceptance Test.

2.3 Cold Cathode Pressure Gauge (PG)

Was not performed since the pressure gauge was not available.

2.4 Transient Pulse Monitor (TPM)

Not requested (see paragraph 4.3.2.3 below).

2.5 Langmuir Probe (LP)

2.5.1 Sweep Voltage

2.5.1.1 Connect LP head simulator to LP electronics (LPE). Note:

Langmuir probe head test was not performed since the probe head has not been assembled to the boom due to uncertainty in carrier requirements.

- 2.5.1.2 Connect LPE output to oscilloscope.
- 2.5.1.3 Turn on LPE and the LP head simulator and measure sweep voltage wave shape and voltage levels.
See Photo #1.
- 2.5.1.4 Compare sweep wave shape and voltage levels with calibrated data.

Data

1. Sweep dropped quickly to -3.0 V.
2. Ramped up to +5.0 V in 0.5 sec.
3. Ramped back down to -3.0 V in 0.5 sec.
4. Rose quickly to +3.0 V and held for 1.0 sec.
5. Dropped to 0.0 V at end of sweep.
6. Total sweep time was 2.0 sec.

Reference: Polaroid taken on 6-21-90 by R. Mueller. H-scale is 0.5 sec/div.; V-scale is 2.0 V/div.

Data Analysis

Data indicates correct operation of the sweep voltage.

Completed - Date: 6-21-90, by _____

- 2.5.2 Electron Density (N_e)
Not requested.
- 2.5.3 Electron Temperature (T)
Not requested.
- 2.6 Emitter
Not requested.
- 2.7 GSE Subsystem
 - 2.7.1 Verify operation and range of all power supplies.
 - 2.7.2 Verify proper voltage on all wafer switches at all switch positions.
 - 2.7.3 Verify functionality of all controls, displays and meters.

Data Analysis

Range and operation of all power supplies was checked and all were working properly. All wafer switch voltages were correct. All controls functioned properly.

Completed - Date: 6-21-90, by _____

- 2.8 PASP Plus Subsystem
- 2.8.1 Array Switching
- 2.8.1.1 Array Selection
- 2.8.1.1.1 Install PASP Plus subsystem in GSE. Connect DACS CARRIER INTERFACE POWER (+28 V) cable into the ASIS chassis instead of the DACS chassis. Attach ASIS Manual Control Unit (AMCU) to Digital ASIS connector on ASIS chassis.
- 2.8.1.1.2 Set the ASIS FUNCTION switch on the AMCU to IV and the ARRAY SELECT switch to Array #1. Record if Array #1 is indicated on the GSE SIM chassis.
- 2.8.1.1.3 Repeat above for Arrays #2 through #6 and then again for the second set of ASIS electronics.

Data

Array #	Selection	
	Set #1	Set #2
1	OK	OK
2	OK	OK
3	OK	OK
4	OK	OK
5	OK	OK
6	OK	OK

Data Analysis

Data shows correct array selection and indication.

Completed - Date: _____, by _____

- 2.8.1.2 Array Frame Grounding (concentrator arrays only)
- 2.8.1.2.1 Make sure the HV SELECT switch on the AMCU is set to 0, then set the ARRAY SELECT switch to Array #3 and the ASIS FUNCTION switch to IV. Set the ARRAY/RTD/HV LOAD selector switch on the CAL chassis to IV. Measurements are made at the connector on the end of the Array Cable coming from the ASIS (for pin numbers refer to drawing #10136777. When the ARRAY SELECT switch is not in a concentrator array position (#3, #4 or #6), then the positive, negative, and frame leads should all be connected together and tied to spacecraft ground. When the ARRAY SELECT switch is set to a concentrator position, then the leads should be connected as follows:
 - 1) In the IV mode the positive lead is open, the negative lead is open (1 megohm to S/C ground), and the frame is open (1 megohm to S/C ground);
 - 2) In the HV mode the positive and negative leads are open and the frame is tied to S/C ground;
 - 3) In the HVT2 mode all three leads are open with the negative and frame leads tied together.
- 2.8.1.2.2 Record if frame lead is properly connected.
- 2.8.1.2.3 Repeat steps 2.8.1.2.1 and 2.8.1.2.2 above for arrays 4

and 6.

Data

Array	Frame Grounding	
<u>#</u>	<u>HV Mode</u>	<u>HVT2 Mode</u>
3	S/C	Neg. lead
4	S/C	Neg. lead
6	S/C	Neg. lead

Data Analysis

Data shows correct frame grounding.

Completed - Date: 6-22-90, by _____

- 2.8.2 IV Curve
- 2.8.2.1 Current Axis
- 2.8.2.1.1 Connect the Array ASIS, Analog ASIS and Emitter ASIS cables from ASIS chassis to CAL chassis. Make sure that the CAL chassis power switch is turned off.
- 2.8.2.1.2 Set the METER FUNCTION selector switch on the CAL chassis of the GSE to IV-I and the ARRAY/RTD/HV LOAD selector switch to Array #1. Set the ARRAY SELECT switch on the AMCU to 1, the ASIS FUNCTION switch to IV and the IV LOAD SELECT switch to F3. Set DMM #1 to measure Amps.
- 2.8.2.1.3 Turn on CAL chassis power switch. When the 0-1 amp current power supply control is set to a particular value, that value will be displayed on DMM #1. The corresponding ASIS analog telemetry output voltage will be displayed on DMM #2.
- 2.8.2.1.4 Set the power supply to the range of interest.
- 2.8.2.1.5 Repeat steps 2.8.2.1.3 and 2.8.2.1.4 above with the ARRAY/RTD/HV LOAD selector switch on the CAL chassis and the ARRAY SELECT switch on the AMCU set to the Array #2 through Array #6 positions. Turn off the CAL chassis power switch.
- 2.8.2.1.6 Compare the applied current values with the corresponding analog telemetry values.

Data

Array	I (mA)	ITELEM(V)	Output
<u>#</u>	<u>Input</u>	<u>Set #1</u>	<u>Set #2</u>
1	300	3.235	3.039
2	250	2.555	2.588
3	350	3.585	3.478
4	500	5.017	5.190
5	350	3.523	3.595
6	350	3.506	3.464

Data Analysis

Input current values were selected to be at the high end of the array output current range. The output voltage for array #4 indicates a need to increase the value of R12 or decrease R13 (preferred) by about 10% on both sets of the IV4 printed circuit board. This will reduce ITELEM by about 10%.

Completed - Date: 6-26-90, by _____

2.8.2.2 Voltage Axis

- 2.8.2.2.1 Set the IV LOAD SELECT switch on the AMCU to 00 and the ARRAY SELECT switch to #1. Set the METER FUNCTION selector switch on the CAL chassis to IV-V and the ARRAY/RTD/HV LOAD selector switch to Array #1. Set DMM #1 to measure Volts.
- 2.8.2.2.2 When the 20-120 volt power supply control is set to a particular value, that value will be displayed on DMM #1. The corresponding ASIS analog telemetry output voltage will be displayed on DMM #2.
- 2.8.2.2.3 Set the power supply to the range of interest and turn on the CAL chassis power switch.
- 2.8.2.2.4 Repeat the above process with the ARRAY/RTD/HV LOAD selector switch and the ARRAY SELECT switch both set to the Array #2 position.
- 2.8.2.2.5 Repeat steps 2.8.2.2.2. through 2.8.2.2.4. for Array #3 through Array #6 (Note: for these tests the 0-20 volt power supply is used in place of the 20-120 volt power supply). Turn off the CAL chassis power switch.
- 2.8.2.2.6 Compare the applied voltage values with the corresponding analog telemetry values.

Data

Array #	V Input	VTELEM(V) Set #1	Output Set #2
1	85	4.988	5.002
2	110	4.442	4.425
3	10	4.517	4.556
4	14	4.765	4.749
5	7	4.867	4.871
6	7	4.873	4.819

Data Analysis

The input voltage values were selected to be at the high end of the array output voltage so the output voltage values should be near the maximum (5 V). The output voltage of array #1 indicates a need to decrease R10 or increase R11 (preferred) by about 10% on both sets of the IV1 printed circuit board. This will reduce VTELEM by about 10%.

Completed - Date: 6-26-90, by _____

- 2.8.3 High Voltage Power Supply
- 2.8.3.1 Voltage Level (ETELEM channel)
- 2.8.3.1.1 Make sure DMM #1 is set to measure Volts. Set the ARRAY SELECT switch on the AMCU to Array #1, the HV SELECT switch to 0 and the ASIS FUNCTION switch to HV. Set the METER FUNCTION selector switch on the CAL chassis to HV-V.
- 2.8.3.1.2 Set the ARRAY/RTD/HV LOAD selector switch to Array #1 position which will apply a 50K ohm load across the high voltage supply. Turn on the CAL chassis power switch.
- 2.8.3.1.3 Step the ASIS high voltage power supply from 0 through +500 volts by selecting voltages with the HV SELECT switch and collect the data of the actual output voltage (DMM #1) and the corresponding ASIS analog telemetry output voltage (DMM #2) across the appropriate high voltage load.
- 2.8.3.1.4 Repeat 2.8.3.1.1 through 2.8.3.1.3 with the ARRAY/RTD/HV LOAD selector switch in the CAL chassis and the ARRAY SELECT switch in the AMCU in the Array #2 through Array #5 positions. This will apply a 25K, 10K, 5K, and 4.5K ohm load respectively across the high voltage supply. Compare the output voltage values with the corresponding analog telemetry values. The output voltage is normally affected slightly with changing load. However, when the Array #5 position is selected (4.5K load) the high voltage supply over-current protection circuitry should disable the high voltage supply (current greater than 100 mA) for two seconds when it is stepped to 500 V. Turn off the CAL chassis power switch.

Data

Load	HV(V)	HV(V) Input		ETELEM(V) Output	
<u>(K)</u>	<u>Level</u>	<u>Set #1</u>	<u>Set #2</u>	<u>Set #1</u>	<u>Set #2</u>
50	50	49.8	49.3	0.253	0.253
	100	99.0	97.4	0.510	0.504
	150	148	146	0.766	0.760
	300	311	308	1.62	1.60
	500	506	501	2.64	2.62
25	50	49.7	49.2	0.253	0.253
	100	98.8	97.3	0.510	0.504
	150	148	146	0.766	0.760
	300	310	307	1.62	1.60
	500	505	500	2.64	2.62
10	50	49.4	48.9	0.253	0.253
	100	98.3	96.7	0.510	0.505
	150	147	145	0.766	0.760
	300	308	305	1.62	1.60
	500	505	497	2.64	2.62
5	50	49.2	48.4	0.253	0.253

	100	97.3	95.7	0.510	0.505
	150	145	144	0.766	0.761
	300	306	302	1.62	1.60
	500	497	494	2.64	2.61
4.5	50	48.9	48.4	0.253	0.253
	100	97.2	95.5	0.510	0.505
	150	145	143	0.766	0.760
	300	305	302	1.62	1.60
	500	---	--Over current--	---	---

Data Analysis

The data shows correct operation of the high voltage power supply and the high voltage analog channels. It is obvious from the data that the high voltage analog circuit is not at 5 V at the high end. This is due to the original request to have a 1000 V power supply capability.

Completed - Date: 6-27-90, by _____

2.8.3.2 Noise

- 2.8.3.2.1 Attach an oscilloscope to the output of the high voltage bias (HVO Board) supply by placing a probe across C18.
- 2.8.3.2.2 Select the first positive bias voltage step value and measure the corresponding peak-to-peak noise.
- 2.8.3.2.3 Repeat above for remaining selected positive bias voltage values. Note: Negative values will not be done since polarity is changed by switching leads.
- 2.8.3.2.4 Compare measured noise with noise specifications.

Data

HV(V)	Noise (V pk-pk)	
<u>Output</u>	<u>Set #1</u>	<u>Set #2</u>
+ 50	1	0.4
+100	2	0.8
+150	3	1.1
+300	4	1.9
+500	2	2.4

References: Set #1 - Polaroid photos taken at +50, +300 and +500 on 6-20-90. See Photos #2-#4. H-scale is 10 msec./div.; V-scale is 0.5 V/div. for #2 and 1.0 V/div. for #3 and #4. Set #2 - Polaroid photos taken at +50, +150, +300 and +500 on 6-26-90. See Photos #5-#8. H-scale is 10 msec./div.; V-scale is 0.5 V/div.

Data Analysis

Noise levels are less than 2% of full scale for all voltages on both sets.

Completed - Date: 6-26-90, by _____

- 2.8.3.3 Polarity
Was not done since voltmeter on SIM chassis of the GSE now measures polarity.
- 2.8.3.4 Leakage Current (CTELEM channel, high leakage current range)
- 2.8.3.4.1 Set the ARRAY SELECT switch on the AMCU to #1, the HV SELECT switch to 0, and the ASIS FUNCTION switch to HV. Set DMM #1 to measure Amps. Set the METER FUNCTION selector switch on the CAL chassis to HV-I.
- 2.8.3.4.2 Set the ARRAY/RTD/HV LOAD selector switch on the CAL chassis to Array #1 which will apply a 50K ohm load across the high voltage supply.
- 2.8.3.4.3 Step the ASIS high voltage power supply from 0 through +500 volts by selecting voltages with the HV SELECT switch on the AMCU and collect the data of the actual leakage current (DMM #1) and the corresponding ASIS analog telemetry output voltage (DMM #2) across the appropriate high voltage load.
- 2.8.3.4.4 Repeat 2.8.3.4.1 through 2.8.3.4.3 for the ARRAY/RTD/HV LOAD selector switch in the CAL chassis and the ARRAY SELECT switch in the AMCU in the Array #2 through Array #5 positions. This will apply a 25K, 10K, 5K, and 4.5K ohm load respectively across the high voltage supply. Compare the output voltage values with the corresponding analog telemetry values. The output voltage is normally affected slightly with changing load. However, when the Array #5 position is selected, the high voltage supply over-current protection circuitry should disable the high voltage supply (current greater than 100 mA) for two seconds when it is stepped to 500 V.
- 2.8.3.4.5 Step the ASIS high voltage power supply and collect the data of actual leakage current (DMM #1) and corresponding analog telemetry output voltage (DMM #2) across the appropriate high voltage load.
- 2.8.3.4.6 Compare the actual leakage current output values with the corresponding analog telemetry values. The output current signal will vary with applied voltage and load resistance. However, when the Array #5 position is selected the high voltage supply over-current protection circuitry should disable the high voltage supply for two seconds when it is stepped to 500 V.

Data

Load HV(V)	I Input (mA)	CTELEM(V)	Output
<u>(K)</u> <u>Level</u>	<u>Set #1</u> <u>Set #2</u>	<u>Set #1</u>	<u>Set #2</u>

50	50	0.995	0.987	0.0377	0.0435
	100	1.98	1.95	0.0794	0.0843
	150	2.96	2.93	0.121	0.126
	300	6.22	6.15	0.259	0.262
	500	10.1	10.0	0.423	0.425
25	50	1.96	1.94	0.0783	0.0836
	100	3.89	3.83	0.160	0.164
	150	5.82	5.75	0.241	0.245
	300	12.2	12.1	0.512	0.513
	500	19.9	19.7	0.835	0.831
10	50	4.95	4.89	0.204	0.208
	100	9.83	9.67	0.410	0.409
	150	14.7	14.5	0.615	0.614
	300	30.9	30.5	1.30	1.29
	500	50.2	49.7	2.11	2.10
5	50	9.80	9.67	0.408	0.409
	100	19.5	19.1	0.815	0.807
	150	29.1	28.7	1.22	1.21
	300	61.1	60.4	2.57	2.55
	500	99.4	98.3	4.19	4.15
4.5	50	10.7	10.6	0.447	0.449
	100	21.3	21.0	0.894	0.885
	150	31.9	31.5	1.34	1.33
	300	67.0	66.2	2.82	2.79
	500	---	--Over current--	---	---

Data Analysis

The data indicates correct operation of the CTELEM analog channel and the over-current protection circuit.

Completed - Date: 6-26-90, by _____

- 2.8.4 Electrometer (ELECT channel, low leakage current range)
- 2.8.4.1 Unplug HVO P3 and HVD P3 plugs from ASIS circuit boards and connect the HVO-P3 JUMPER plug to HVO P3.
- 2.8.4.2 Set the ARRAY SELECT switch on the AMCU to Array 6 and the ASIS FUNCTION switch to HV. Set the METER FUNCTION selector switch on the CAL chassis to ELECT and the ARRAY/RTD/HV LOAD selector switch to Array 6/Elect. Set DMM #1 to measure Amps.
- 2.8.4.3 The input current determined by the setting of the ELECTROMETER CURRENT selector switch will be indicated on DMM #1. The corresponding ASIS analog telemetry output voltage will be indicated on DMM #2.
- 2.8.4.4 Step the ELECTROMETER CURRENT selector switch through the full range of values as the controller collects the data.
- 2.8.4.5 Remove the jumper plug from HVO J3 and reconnect HVO P1 and HVD P1 plugs.
- 2.8.4.6 Compare the applied current values with the corresponding analog telemetry values.

Data

Switch Setting	I Input (uA)		ELECT(V) Output	
	Set #1	Set #2	Set #1	Set #2
10nA	---	---	+5.006*	+5.25*
30nA	---	---	+5.006*	+5.25*
100nA	---	---	+5.006*	+5.25*
300nA	0.317	0.318	+1.326	+1.25
1uA	1.062	1.062	+0.9379	+0.9302
3uA	3.182	3.180	+0.7427	+0.7400
10uA	10.33	10.32	+0.5555	+0.5550
30uA	31.44	31.44	+0.3849	+0.3850
100uA	99.42	99.43	+0.2108	+0.2114
300uA	306.9	306.9	+0.0423	+0.0433
1mA	933.0	933.0	-0.0735	-0.0704

* Saturated

Data Analysis

The ELECT analog channel is working properly. Input voltage to the log amplifier in the electrometer actually swings slightly negative somewhere below 300nA and causes the electrometer to saturate. This was necessary to provide maximum sensitivity down to 300nA before the noise threshold takes over.

Completed - Date: 6-26-90, by _____

- 2.8.5 Emitter (EMITTER channel)
- 2.8.5.1 Set ARRAY SELECT switch on the AMCU to Off and the ASIS FUNCTION switch to Off. Set the METER FUNCTION selector switch on the CAL chassis to EMITTER and the ARRAY/RTD/HV LOAD selector switch to Off. Set DMM #1 to measure Amps.
- 2.8.5.2 When the emission current, as determined by the 0-100 mA EMISSION power supply, is set to a desired value the result will be indicated on DMM #1. The corresponding ASIS analog telemetry voltage output will be indicated on DMM #2.
- 2.8.5.3 Set the EMITTER emission current power supply to the range of interest and turn on the CAL chassis power switch. After data has been collected turn off the CAL chassis power switch.
- 2.8.5.4 Compare the applied current values with the corresponding analog telemetry values.

Data

Current(mA) Input	EMITTER Output (V)	
	Set #1	Set #2
100	1.0004	0.9997
80	0.8001	0.7998

50	0.4990	0.4999
20	0.1983	0.2000

Data Analysis

The EMITTER channel of the ASIS is functioning properly.

Completed - Date: 6-28-90, by _____

- 2.8.6 RTD Temperature Sensors
- 2.8.6.1 Set the ARRAY/RTD/HV LOAD selector switch to RTD #1.
- 2.8.6.2 Set the RTD SIMULATOR selector switch to the lowest resistance value (30 ohms). Turn on the CAL chassis power switch.
- 2.8.6.3 Turn on the GSE controller. The controller will boot up to an overhead menu bar screen in the FPC subdirectory. Place a floppy disk in the B: drive. Hit Escape to get out of the menu and then type ' and a space to get into DOS. Then type B: and Return to select the B drive as the default drive. Get back into the menu program by typing tt and then Return. Select the Operations (O) menu then the Dacs commands (D) menu, then turn the Dacs power ON (D), and then Quit (Q).
- 2.8.6.4 Select the Cal commands (C) menu and then Rtd data packet (R), and then Quit (Q).
- 2.8.6.5 Select the Telemetry (T) menu. Toggle the View capability to ON. Select Begin (B) which will then indicate a "WAITING" state. When the telemetry state shifts to "RUNNING" select Quit (Q) and immediately select the Gse test commands (G) menu and then the Abbreviated test sequence (A). Note: when telemetry is running data is being collected and stored on the floppy disk under a file named rdo-00.cap. The longer the time between starting the telemetry and starting the test sequence, the larger the amount of useless data there will be in the beginning of the capture file. The same is true at the end of a sequence with regard to turning off the telemetry.
- 2.8.6.6 The abbreviated test sequence would normally run for about 3.7 minutes but the desired RTD data is contained in the I-V curve portion at the start of the test. After the I-V curve data collection is completed (when the series of rapid relay clicks stops) wait for about 12 seconds for the data to pass through the buffers and be written to the disk file. At this time select Quit (Q) and immediately select Telemetry (T) and then End (E). There will be a pause after selecting End while the buffer is being copied to the disk. After this pause select Quit (Q), then Dacs commands (D), and then Warm restart (W). Remove the floppy disk and rename the capture file rdo-00.cap as CTS12.CAP.
- 2.8.6.7 Repeat the above process with the ARRAY/RTD/HV LOAD

selector switch set to the RTD #2 through RTD #6 and then RTD #1 and #2 positions again while selecting successively higher resistances to 170 ohms. The files should be labeled CTS13 through CTS19 respectively. At the end of the tests turn the CAL chassis power switch off.

- 2.8.6.8 Compare the selected resistance values with the corresponding digital telemetry values in the appropriate array data sequences and the appropriate RTD calibration tables.

Data

Array #	RTD Resist.	Raw Data Reading	Calculated Value	Temp. °C	Tabular Value °C
1	30	B3 03	947	-178.8	-173.1
2	50	33 03	819	-126.2	-125.2
3	70	BE 02	702	- 78.2	- 75.8
4	90	3D 02	573	- 25.2	- 25.5
5	110	BF 01	447	+ 26.5	+ 25.7
6	130	48 01	328	+ 75.3	+ 77.7
1	150	CE 00	206	+125.4	+130.5
2	170	4B 00	75	+179.2	+184.2

Data Analysis

The RTD temperature scale is set up as a linear scale from +210 down to -210 °C and uses hexadecimal values from 0 (at -5 V) to 3FF (at +5 V) respectively to represent the voltage outputs from the RTDs at those temperatures. The most significant byte is obviously the second byte in the hexadecimal raw data with 3FF (1023 DEC) as the highest value. The tabular values are derived by using the indicated calibration resistance values as if they were actual RTD gauge resistance values and then consulting a table based on DIN 43760 values.

Completed - Date: _____, by _____

- 2.8.7 Analog/Digital Conversion
- 2.8.7.1 Set the METER FUNCTION selector switch to AD.
- 2.8.7.2 Initiate the measurement process by following the procedure in sections 2.8.6.3 and 2.8.6.5 above except that the Cal menu ad1 mux a data packet (Alt-F1) for channel 4 should be selected and just before starting the telemetry manually set the -5v/+5v power supply to the first desired voltage level (-4.99 V). The actual applied voltage will be indicated on DMM #1 and the digital telemetry voltage will be captured on a floppy disk capture file which should be named CTS1.CAP. The data of interest will be found in the V- and I-channels

- of arrays 1 and 4.
- 2.8.7.3 Perform 2.8.7.2 and 2.8.7.3 for ten other selected voltages except the capture files should be named CTS2 through CTS11. While this test only checks four of the 32 A/D channels, the channels that are tested are located in each of the two multiplexer circuits on each of the two AD circuit boards. This then serves as a hardware check of the A/D convertor but not as a check of each of the input channels' gain or wiring.
- 2.8.7.4 Compare the actual applied voltage values with the corresponding digital telemetry values.

Data

<u>Voltage Setting</u>	<u>Raw Data* Values (HEX)</u>	<u>Calculated* Values (DEC)</u>
-4.9825	01-02	1-2
-3.9996	34-35	52-53
-2.9978	67-68	103-104
-1.9982	9A-9B	154-155
-0.9995	CC-CD	204-205
-0.0003	01-FF	0 or 255
+0.9982	33-34	51-52
+1.9996	66-67	102-103
+2.9995	99-9A	153-154
+3.9985	CC-CD	204-205
+4.9822	FE-FF	254-255

Data Analysis

*Raw data bits occasionally shifted during collection of the 32 data points. Note also that negative values are handled using a bitwise complement scheme so that twice as much sensitivity is available in one data byte.

Completed - Date: _____, by _____

- 2.8.8 Sun Sensor Interface
- 2.8.8.1 Connect the sun sensor simulator in the GSE to the DACS.
- 2.8.8.2 Select the Setup (S) menu, then the Sun sensor simulator (S), then 1 for the X-axis and enter 20 (will show as 14 HEX on the simulator chassis panel). Select 3 for the Y-axis and then enter 20 (14 HEX). Finally toggle automatic threshold adjust (Ata) to ON. Quit and then select Telemetry (T) which will present a new menu. Toggle the View capability to ON. Select Begin (B) which will then indicate a "WAITING" state. When the telemetry state shifts to "RUNNING" select Quit (Q) and immediately move cursor to the left to the Operations menu where Gse test commands (G) menu is selected and then the Abbreviated test sequence (A) should be selected.
- 2.8.8.3 The abbreviated test sequence would normally run for

about 3.7 minutes but the desired sun sensor data is contained in bytes 7-9 after each of the six IV data markers at the start of the test. After the I-V curve data collection is completed wait for about 12 seconds for the data to pass through the buffers and be written to the disk file. At this time select Quit (Q) and immediately select the Telemetry (T) menu and then End (E). There will be a pause after selecting End while the buffer is being copied to the disk. Remove the floppy disk and rename the file CTS20.CAP.

- 2.8.8.4 Repeat above with 40 (28 HEX), 40 (28 HEX), OFF data from simulator and mark the file CTS21.CAP.
- 2.8.8.5 Compare simulated data to digital telemetry data.

Data

<u>SS Setting</u>	<u>Raw Data</u>	<u>Corrected Data</u>
20, 20, ON	18 81 FE	14, 14, FE
40, 40, OFF	24 42 FF	28, 28, FF

Note: The SSI circuit board has a wiring flaw which swaps the 4 and 8 bits and prior to 9-5-90 the sun sensor simulator wiring swapped the two nibbles on the Y-axis data. On 9-5-90 the nibble wiring was corrected. The nibble swapping only occurred when the simulator was used while the 4 and 8 bit swapping occurs for all raw sun sensor data. Any data taken when the automatic threshold adjust is off may not be correct. The data shown above as derived in the OFF mode was actually taken in the ON mode except for the FF signal for OFF.

Completed - Date: _____, by _____

- 2.8.9 Pressure Sensor
Not requested.
- 2.8.10 Housekeeping Data
 - 2.8.10.1 Power Supply Voltages
 - 2.8.10.1.1 Remove voltage sensing plug (P2) from the house-keeping circuit board in the ASIS electronics (HK1).
 - 2.8.10.1.2 Apply low and high test voltages to the window circuit of channel 0 on the HK1 board while collecting housekeeping data (digital data from optical isolators to J1). See Table 2.8.10.1-1 below for ideal test voltages.

Table 2.8.10.1-1 - Housekeeping Voltage Ranges

HK1, Connector J-2				Voltage Range	
Chan	Pin	Num.	Power		
Num.	+	-	Supply	Nom.	High Low

0	2	21	+5V EMIT	5.0	5.5	4.5
1	4	23	+5V ASIS	5.0	5.5	4.5
2	6	25	+15V ASIS	15.0	16.5	13.5
3	27	8	-15V ASIS	15.0	16.5	13.5
4	10	29	+24V RLY	24.0	26.4	21.6
5	12	31	+15VAD210	15.0	16.5	13.5
6	33	14	-15V HVE	15.0	16.5	13.5
7	16	35	+15V HVE	15.0	16.5	13.5

HK2, Connector J-2

Chan Num.	Pin +	Num. -	Power Supply	Voltage Range		
				Nom.	High	Low
0	2	21	+5V AK	5.0	5.5	4.5
1	4	23	+5V MO	5.0	5.5	4.5
2	6	25	+5V LO	5.0	5.5	4.5
3	8	27	+5V DAC	5.0	5.5	4.5
4	10	29	+15V DAC	15.0	16.5	13.5
5	31	12	-15V DAC	15.0	16.5	13.5
6	14	33	+28V SW	28.0	30.8	25.2
7	16	35	+5V SPARE	5.0	5.5	4.5

- 2.8.10.1.3 Repeat above for channels 1 through 7 of HK1.
 2.8.10.1.4 Compare applied low and high window voltages with ideal window voltages as shown in Table 2 above.
 2.8.10.1.5 Repeat 2.8.10.1.1 through 2.8.10.1.4 for the HK2 board located in the DACS.

Data

Set #1

HK1, Connector J-2

Chan Num.	Pin +	Num. -	Power Supply	Voltage Range		
				Nom.	High	Low
0	2	21	+5V EMIT	4.98	5.49	4.46
1	4	23	+5V ASIS	5.04	5.57	4.51
2	6	25	+15V ASIS	15.06	16.71	13.41
3	27	8	-15V ASIS	15.07	16.73	13.40
4	10	29	+24V RLY	24.15	26.85	21.45
5	12	31	+15VAD210	15.09	16.74	13.44
6	33	14	-15V HVE	14.97	16.62	13.31
7	16	35	+15V HVE	14.94	16.60	13.28

HK2, Connector J-2

Chan Num.	Pin +	Num. -	Power Supply	Voltage Range		
				Nom.	High	Low
0	2	21	+5V AK	4.98	5.47	4.48
1	4	23	+5V MO	5.04	5.56	4.52
2	6	25	+5V LO	4.98	5.48	4.47
3	8	27	+5V DAC	4.99	5.51	4.47
4	10	29	+15V DAC	14.95	16.64	13.26
5	31	12	-15V DAC	15.08	16.74	13.42
6	14	33	+28V SW	28.11	31.36	24.86
7	16	35	+5V SPARE	4.96	5.46	4.46

Set #2

HK1, Connector J-2

Chan Num.	Pin	Num.	Power Supply	Voltage Range		
				Nom.	High	Low
0	2	21	+5V EMIT	5.03	5.54	4.51
1	4	23	+5V ASIS	4.97	5.47	4.46
2	6	25	+15V ASIS	15.04	16.66	13.12
3	27	8	-15V ASIS	15.00	16.61	13.39
4	10	29	+24V RLY	24.10	26.73	21.47
5	12	31	+15VAD210	15.08	16.75	13.41
6	33	14	-15V HVE	15.05	16.68	13.42
7	16	35	+15V HVE	15.09	16.72	13.46

HK2, Connector J-2

Chan Num.	Pin	Num.	Power Supply	Voltage Range		
				Nom.	High	Low
0	2	21	+5V AK	5.01	5.52	4.50
1	4	23	+5V MO	5.04	5.55	4.52
2	6	25	+5V LO	5.00	5.51	4.49
3	8	27	+5V DAC	5.03	5.54	4.52
4	10	29	+15V DAC	14.94	16.56	13.32
5	31	12	-15V DAC	15.05	16.68	13.41
6	14	33	+28V SW	27.96	31.00	24.92
7	16	35	+5V SPARE	5.02	5.53	4.51

Data Analysis

The housekeeping window circuits are adjusted to a value near the nominal value in all cases.

Completed - Date: _____, by _____

2.8.10.2 Housekeeping Temperatures
Not requested.

2.8.11 DACS Commands

2.8.11.1 Response

2.8.11.1.1 Run through entire set of DACS system commands by selecting the Operations (O) menu and then the Dacs commands (D) menu and then each individual DACS command.

2.8.11.1.2 Compare actual PASP Plus system response to commands with expected response.

Data

Each DACS command was exercised and the resultant response was identical to the expected response except the ORBIT 6 command termination was not clean in that it did not result in a shift to low power or give some other indication of completion.

Completed - Date: _____, by _____

2.8.11.2 Sequence

Use data generated in step 2.8.11.1.1 above and compare actual event sequence and timing with expected events and timing.

Data

Each DACS sequence command was exercised and the resultant sequence was identical to the expected sequence. The length of one DAE was found to be about 15 minutes, 39 seconds long. Pauses between DAE #1/DAE #2 and DAE #2/DAE #3 were about 1 minute, 15 seconds long which, along with a 34 minute, 25 second occultation period, gave an orbit sequence time of 83 minutes, 52 seconds.

Completed - Date: _____, by _____

2.8.12 Transient Pulse Monitor (TPM) (Note: this test was superseded by acceptance test performed at JPL with SRI personnel - see section 4.3.2.3 below)

2.8.12.1 Commands

Use data obtained in step 2.8.11.1.1 above to see if all TPM commands were properly received and decoded by the transient pulse monitor interface board (TPI).

2.8.12.2 Buffering

Use data from 2.8.11.1.1 above to check on buffering of TPM data and inclusion in DACS to spacecraft data stream.

Data

TPM data is included in normal data stream but the buffering of arc events was not tested.

2.8.13 Telemetry

Use data from 2.8.11.1.1 above to check formatting of DACS telemetry data versus specified data format.

Data

The telemetry data is formatted into packets of 512 bytes. The start of each packet contains 28 bytes of non-data information. The primary header takes up the first six bytes, the secondary header takes up the next ten bytes, the ancilliary data (sun sensor data and unused data) takes up eight bytes, and pointers take up the last four bytes. At the end of each packet is the cyclic redundancy check (CRC) which takes up the last two bytes leaving 482 bytes for data markers, counts and data. The data dictionary included in the software

description (Appendix V) gives sufficient detail so that the significance of each byte can be determined.

Completed - Date: _____, by _____

- 2.8.14 Subsystem Thermal Test (this test was not performed)
The overall plan of this portion of the test procedure was to run a baseline test using an abbreviated, computer-controlled test sequence with simulated input signals for each sensor at room temperature (+25 °C) and then run verification tests at low and high system operating temperatures. Examining the outputs obtained from these tests would disclose the output differences made by the temperature changes. The abbreviated test sequence contains all of the operating commands and data collection of a normal flight data acquisition element sequence (see Figures 1 and 2) except that the 20 second application of each of the voltage bias steps has been reduced to about two seconds. This allows a 15 minute, 39 second normal flight data acquisition element sequence to be run in about 3.7 minutes.
- 2.8.14.1 Make sure that the EPROMs are loaded on the SEQ board of the DACS.
 - 2.8.14.2 Place ASIS and DACS electronics from the GSE into a temperature chamber. First, an abbreviated test sequence at ambient temperature (+25 °C) will be run.
 - 2.8.14.3 Place a floppy disk in the B: drive and turn on the GSE controller. The controller will boot up to an overhead menu bar screen. Select the Operations (O) menu then the Dacs commands (D) menu, then turn the Dacs power ON (D), and then Quit (Q).
 - 2.8.14.4 Select the Setup (S) menu, then Telemetry (T) which will present a new menu. Toggle the View capability to ON. Select Begin (B) which will then indicate a "WAITING" state. When the telemetry state shifts to "RUNNING" select Quit (Q) and immediately move cursor to the left to the Operations menu where the Gse test commands (G) menu is selected and then the Abbreviated Test Sequence (A) should be selected.
 - 2.8.14.5 The abbreviated test sequence will run for about 3.7 minutes - wait for the emitter light to go out after high voltage testing of array 6 is completed. At the end of the test select Quit (Q) and then cursor over to the setup menu and select Telemetry (T) and then End (E). There will be a pause after selecting end while the buffer is being copied to the disk. Remove the floppy disk and rename the file ATS1.CAP.
 - 2.8.14.6 Gradually lower test chamber temperature to 0 °C and repeat above except capture file is named ATS2.
 - 2.8.14.7 Gradually raise test chamber temperature to +50 °C and

- repeat above except capture file is named ATS3.
- 2.8.14.8 Compare test results with previous data and then remove equipment from chamber.
 - 2.8.14.9 Repeat steps 2.8.14.1 through 2.8.14.7 except use brassboard ASIS and DACS electronics and use capture files named ATS4, ATS5, and ATS6.

3 Interface Test Results

3.1 Mechanical

- 3.1.1 Check for presence and proper location of all system components.
- 3.1.2 Check for proper mounting of all system components.
Note: unless specifically required all mounting hardware is to be only finger tight during system tests so as to prevent galling or damage to equipment.

Data

All mechanical checks were made and there were no problems except it was found that mounting the sun sensor head on a bench allowed best control of test conditions.

Completed - Date: _____, by _____

3.2 Connectors

3.2.1 General

- 3.2.1.1 Check connectors for presence of proper labelling before mating.
- 3.2.1.2 Visually inspect all connector pins for straightness, alignment and contamination.

Data

All connectors and cables were inspected and found to be in good condition. Proper procedure would be to clean the connector pins just prior to shipment but this was not done due to lack of resources.

Completed - Date: _____, by _____

3.2.2 Power Connectors

- 3.2.2.1 Before mating power connectors check voltage level and polarity and record results on a log sheet.
- 3.2.2.2 Compare logged values with correct power voltage conditions and then initial and date log sheet unless values do not coincide.

Data

All power connectors were checked and measured values were the same as expected values.

Completed - Date: 6-28-90, by _____

3.2.3 Signal Connectors

Check continuity and pin assignment against schematic and cabling diagram data.

Data

All cables were checked against schematics and cabling diagrams and continuity and pin assignments were correct.

Completed - Date: 6-28-90, by _____

4 System Test Results

4.1 GSE-Based Operation Tests

The overall plan of the GSE-based operation portion of the baseline test procedure is to run a baseline test using a computer-controlled test sequence with low level or no simulated input signals for each sensor and then run a verification test with high level simulated input signals. Examining the outputs obtained from these two tests will disclose the output differences made by the simulated inputs. The computer-controlled test sequence contains all of the operating commands and data collection of a normal flight orbit sequence (see Figures 1 and 2) and runs for about 3.7 minutes.

4.1.1 Test Setup

4.1.1.1 Wire system per block diagram 10125207. If abbreviated tests sequences are to be run make sure that the TSS plug is in place. If normal length test sequences are to be run use the TST plug.

4.1.1.2 Make sure that the test software is loaded on the PROMs of the SEQ board in the DACS.

4.1.1.3 Place all arc lamp switches in the OFF position.

4.1.1.4 Place Pressure Gauge (PG) simulators for pressure and temperature in the low position.

4.1.1.5 The TPM simulator will be in its initial condition.

4.1.1.6 Set the array simulator SUN/DARK switch to DARK.

4.1.2 GSE Baseline Test (Orbit Test Sequence #1)

4.1.2.1 Turn on the GSE controller. The controller will boot up to an overhead menu bar screen. Place a floppy disk in the B: drive. Hit Escape to get out of the menu and then type ` and a space to get into DOS. Then type B: and Return to select the B drive as the default drive. Get back into the menu program by typing tt and then Return. Select the Operations (O) menu then the Dacs commands (D) menu, then turn the Dacs power ON (D), and then Quit (Q).

4.1.2.2 Select the Setup (S) menu, then the Sun sensor simulator (S), then 1 for the X-axis and enter 20 (will show as 14 HEX). Select 3 for the Y-axis and then enter 20 (14

HEX). Finally toggle automatic threshold adjust (Ata) to ON. Quit and then select Telemetry (T) which will present a new menu. Toggle the View capability to ON. Select Begin (B) which will then indicate a "WAITING" state. When the telemetry state shifts to "RUNNING" select Quit (Q) and immediately select the Gse test commands (G) menu and then the Abbreviated test sequence (A). Note: when telemetry is running data is being collected and stored on the floppy disk under a file named rdo-00.cap. The longer the time between starting the telemetry and starting the test sequence, the larger the amount of useless data there will be in the beginning of the capture file.

- 4.1.2.3 The abbreviated test sequence will run for about 3.7 minutes - wait for the emitter light to go out after high voltage testing of array 6 is completed. At the end of the test, select Quit (Q) and then immediately select Telemetry (T), wait about 12 seconds for the buffers to empty, and then select End (E). There will be a pause after selecting end while the buffer is being copied to the disk. Remove the floppy disk and rename the rdo-00.cap capture file OTS1.CAP.

Completed - Date: _____, by _____

- 4.1.2.4 Test Outputs
- 4.1.2.4.1 AR1 through AR6 simulator IV curve current and voltage data and array temperatures for occultation phase operation as seen through IV1 through IV6 printed circuit board sensors and DACS AD convertor channels.
- 4.1.2.4.2 Array bias voltages.
- 4.1.2.4.3 Plasma leakage current for each array bias voltage.
- 4.1.2.4.4 PG pressure and temperature data for low level input voltages.
- 4.1.2.4.5 TPM channel number, event register count and pulse characterization data for E-field and current sensor channels.
- 4.1.2.4.6 Sun sensor gray code reading for X- and Y-axes equal to 14 HEX (Note: this does not equal the 20 degrees input since gray code has not been allowed for and the 4 and 8 bits are swapped. The indicated angle for 14 HEX would be -51.3 degrees) and an automatic threshold adjust setting of ON.

Data

Digital <u>Command</u>	Desired <u>Voltage</u>	Raw <u>Data</u>	Calc. <u>Data</u>
00	0	FF	255
03	50	0D	13
06	100	1A	26
09	150	27	39

12	300	4E	78
1F	500	7D	125
00	0	06	6
09	150	26	38
0C	200	34-35	52-53
12	300	4E	78
1F	500	7D	125

Data Analysis

Any analysis of the test data in real time was not possible at the time of the tests since there is no way to capture the real-time data flow. An additional problem with data analysis was that there was no funding for post-flight data analysis so the analysis had to be performed manually rather than with the use of computer data reduction. With these caveats in mind, the post-test analysis of the data contained on the OTS1.CAP file showed that the sun sensor simulator threshold was set too high so no data was getting transmitted and that some array simulator outputs were too high and were saturating their respective A/D converter channels over a portion of the I-V curve. These results are reflected in all of the test files generated before 7-17-90. After this date the sun sensor simulator was modified to reduce the threshold level and the sun sensor data was properly transmitted. The saturation of the A/D channels was not addressed since all of this hardware would likely be modified for use with new flight solar arrays. ETELEM data was present for all data points and resulted in a linear plot except that the 500 V point was low. This could be an artifact of the approximation of 16.7 V per step. The ITELEM and ELECT data was not meaningful since no leakage was simulated.

Completed - Date: _____, by _____

- 4.1.3 GSE Verification Test (Orbit Test Sequence #2)
- 4.1.3.1 Test Condition Modifications
- 4.1.3.1.1 Set +230 V and -500 V arc lamp switches to ON.
- 4.1.3.1.2 Set PG simulators to high mode for both pressure and temperature.
- 4.1.3.1.3 Set array simulator SUN/DARK switch to SUN.
- 4.1.3.1.4 Set sun sensor simulator to 40 (28 HEX), 40 (28 HEX), OFF.
- 4.1.3.1.5 Repeat section 4.1.2.2 and 4.1.2.3 except the completed capture file should be renamed OTS2.CAP. Note: After the arc lamps have fired once or twice they should be turned off since their continual arcing has affected some data records.

Completed - Date: _____, by _____

- 4.1.3.2 Test Outputs
 - 4.1.3.2.1 AR1 through AR6 simulator IV curve data. No array temperature data is generated by simulators.
 - 4.1.3.2.2 Array bias voltages with over-current events at +300 and -500 V.
 - 4.1.3.2.3 PG pressure and temperature data for high level input voltages.
 - 4.1.3.2.4 TPM channel number, event register count and pulse characterization data .
 - 4.1.3.2.5 Sun sensor reading for X- and Y-axes equal to 28 HEX and an automatic threshold adjust setting of OFF.
- 4.1.3.3 Test Data Analysis
 - Compare data with similar data obtained in PASF Plus subsystem tests (see step 2.8).

Data Analysis

Post-test data analysis gave the following results:

1. The I-V curves were being produced for both low and high light level tests (low level data from OTS1.CAP) and the resultant curves for Array 1 are included in Figure 2.3.1.2.2-1 of the final report.
2. Over-current events were not done since the arc tube design is incomplete.
3. PG pressure and temperature data was not taken since the PG instrument was dropped.
4. TPM data taken before 7-17-90 was unusable due to a software problem.
5. Sun sensor readings were appropriate for input values if the Y-axis nibbles are swapped and the 4 and 8 bits are swapped for both X- and Y-axes.

Completed - Date: _____, by _____

4.2 PASP Brassboard Tests

The PASP brassboard system tests will exercise all of the functions called out in paragraph 4.1 above. The only difference is the use of the ASIS and DACS brassboard electronics mounted on the system demonstration cart in place of the built-in GSE electronics.

4.2.1 Test Setup

System wiring per block diagram 10125207. Note: the ASIS and DACS electronics along with the pressure sensors will be located on the demonstration test cart.

Completed - Date: _____, by _____

4.2.2 Test Conditions

Repeat paragraphs 4.1.1.2 through 4.1.3.3 above except the data capture files are to be named OTS3.CAP and

OTS4.CAP.

Data Analysis

Post-test data analysis was not as extensive as that done for the GSE hardware. The I-V data was visually scanned to verify operational similarity to the GSE hardware. It was noted that the RTD data taken at the beginning of the IV curve data was incorrect so the RTD value at the end of the IV curve was used for all subsequent data collection. The cause of the poor first data points was probably due to too short an interval between turning on the IV board and reading the data.

Completed - Date: _____, by _____

4.3 System Level Equipment Tests

The system level baseline tests exercise all peripheral arrays and sensors and all channels of the PASP Plus Diagnostics electronics. However, some sensor inputs will be simulated rather than stimulated since certain environmental inputs are difficult or expensive to obtain at a system test level. These simulated inputs will be identified in the detailed test procedures below.

The overall plan of the flight equipment system tests is to run a baseline test using an abbreviated, computer-controlled test sequence with low level or no simulated input signals for each sensor. After the baseline test another test sequence will be run with sensors stimulated or simulated. Examination of the outputs from the two test sequences allows the determination of changes in the output data caused by changes in the inputs of each channel. The numerical data that is obtained will be qualitative in most cases, not quantitative. There are two reasons for this lack of quantitative data: first, the inputs from various stimulators such as floodlights are not controlled or referenced; second, the data has not been subsequently processed using calibration and transfer function tables in order to obtain actual values. This is only a small loss since the actual values of the array and sensor outputs from known inputs have previously been well characterized during acceptance tests performed at JPL or at Air Force facilities or from earlier subsystem testing. This earlier testing along with the results of calibration tests should provide adequate assurance of proper operation of the electronic systems.

4.3.1 Array Tests

4.3.1.1 Array Operation

4.3.1.1.1 Wire system per block diagram 10125209.

- 4.3.1.1.2 Assemble PASP Plus brassboard electronics and flight equipment on system demonstration cart. Note: If actual array mounting is to be simulated use the array mounting torque values shown below.

Table 4.3.1.1-1 - Array Mounting Torques

SP Array	40 in-lbs.
GP Array	40 in-lbs.
CC Array	25 in-lbs.
SC Array	110 in-lbs.

- 4.3.1.1.3 Make sure that the abbreviated test software is loaded on the EPROMs of the SEQ board in the DACS.
- 4.3.1.1.4 Turn on the GSE controller and place a floppy disk in the B: drive as in section 4.1.2.1 above.
- 4.3.1.1.5 Remove protective covers from arrays. Turn on RACK and S/C power.
- 4.3.1.1.6 Turn on floodlights.
- 4.3.1.1.7 Immediately after floodlights are turned on repeat test sequence in sections 4.1.2.2 and 4.1.2.3 above except do not set the sun sensor and name the capture file ATS7.CAP.
- 4.3.1.1.8 Array #1 through #6 IV curve data at about one-fourth sun will be obtained by the controller and written into the capture file along with the other test data.
- 4.3.1.1.9 Turn off floodlights and replace array protective covers.
- 4.3.1.1.10 Turn floodlights back on. Move a reference solar cell over the area of each array and measure the open circuit voltage of the reference solar cell to obtain approximate light level for each array. Record light levels. Turn off floodlights.
- 4.3.1.1.11 Compare test data with acceptance test or subsystem test data after making corrections for light level.

Data

Reference cell readings:

Center of SP array (#1):	28.2 mA
Center of GP array (#2):	25.0 mA
Center of CC array (#3):	45.6 mA
Center of SC array (#4):	32.3 mA

Reference Cell Number XX001 reads 1.083 mA/mW/cm² at AM01.5.

Data Analysis

The reference cell shows light at the center of the arrays at a level of about one-fourth AM0 but plotted data for planar arrays shows data at about

one-fifth for planar arrays and somewhat less for concentrator arrays. There are two reasons for these results: first, the array simulators are set to give worst case (highest) curve values; second, the light intensity varies widely over the arrays and the light is non-collimated which affects the concentrators more than the planars.

Completed - Date: _____, by _____

- 4.3.1.2 Array Isolation
- 4.3.1.2.1 Data obtained in 4.3.1.1.8 contains high voltage bias (ETELEM) and array (or plasma) leakage (ITELEM & ELECT) data for Arrays #1 through #6.
- 4.3.1.2.2 Compare test data with acceptance test or subsystem test data.

Data

Data from tests run before 7-30-90 did not have ETELEM results. After 7-30-90 a broken lead was tracked down in a connector in the high voltage feedback circuit. Once this wire was repaired the ETELEM data was good except for one or two random bad data points. This random bad data may be due to an interaction between the different hardware and the computer timing. ITELEM and ELECT data was expected to be zero and that was what was shown.

Completed - Date: _____, by _____

- 4.3.1.3 Thermal
- 4.3.1.3.1 Rerun test per steps 4.3.1.1.4 through 4.3.1.1.7 except set up capture file named ATS8.CAP and wait five minutes after floodlights are turned on before running test.
- 4.3.1.3.2 Turn off floodlights and replace array protective covers.
- 4.3.1.3.3 Data obtained in 4.3.1.1.8 should contain initial temperature (RTD) data for Arrays #1 through #6 at a level slightly above room temperature.
- 4.3.1.3.4 Data obtained in 4.3.1.3.1 should contain temperature data for Arrays #1 through #6 at a level higher than the previous test.
- 4.3.1.3.5 Compare both sets of test data with acceptance test or subsystem test data. Note: I-V curve data may also be slightly affected by the temperature change.

Data

Array temperatures changed as indicated below:

Array	ATS7 Test Data	ATS8 Test Data
-------	----------------	----------------

#	Raw	Cal.	Temp. °C	Raw	Cal.	Temp. °C
1	C5 01	453	24.0	BB 01	443	28.1
2	C9 01	457	22.4	BA 01	442	28.5
3	C8 01	456	22.8	B7 01	439	29.8
4	D0 01	464	19.5	A3 01	419	38.0
5	00 00	---	----	00 00	---	----
6	00 00	---	----	00 00	---	----

The four GFE arrays all heated up during the five minute wait before running the second test. The two planar arrays and the mini-Cassegrainian concentrator array heated similarly. The SLATS array showed a more extreme temperature change which may indicate the lights were focussed upon the temperature sensor or else the sensor or the heat sink may be defective.

Completed - Date: _____, by _____

4.3.2 Sensor Tests

4.3.2.1 Sun Sensor

- 4.3.2.1.1 Place spotlight about one foot from sun sensor head and align as close to normal with sun sensor head as possible.
- 4.3.2.1.2 Remove sun sensor head protective cover.
- 4.3.2.1.3 Turn on spotlight.
- 4.3.2.1.4 Run abbreviated sequence tests per sections 4.1.2.2 and 4.1.2.3 above except name capture file ATS9.CAP.
- 4.3.2.1.5 Move spotlight so it is aligned as near as possible to 15° from both X and Y axes.
- 4.3.2.1.6 Run abbreviated sequence tests per sections 4.1.2.2 and 4.1.2.3 above except name capture file ATS10.CAP.
- 4.3.2.1.7 Turn off spotlight and replace sun sensor head protective cover.
- 4.3.2.1.8 Data obtained in step 4.3.2.1.4 above contains sun sensor X- and Y-axis outputs near zero degrees while data obtained from 4.3.2.1.6 contains sun sensor data near 15°.
- 4.3.2.1.9 Compare data from both tests with acceptance test or subsystem test data.

Data

The sun sensor data showed appropriate changes when the sensor head is moved. Calibration was not attempted.

Completed - Date: _____, by _____

- 4.3.2.2 Cold Cathode Pressure Gauge (PG) (Was not performed - no PG available)
- 4.3.2.2.1 Disconnect PG sensor head from PGE and connect GSE PG

simulator.

- 4.3.2.2.2 Set PG simulator switches to low level for both pressure and temperature channels.
- 4.3.2.2.3 Run abbreviated test sequence per sections 4.1.2.2 and 4.1.2.3 above except name capture file ATS11.
- 4.3.2.2.4 Set PG simulator switches to high level for both pressure and temperature channels.
- 4.3.2.2.5 Run abbreviated test sequence per sections 4.1.2.2 and 4.1.2.3 above except name capture file ATS12.
- 4.3.2.2.6 Compare data from both tests with acceptance test or subsystem test data.
- 4.3.2.2.7 Reconnect PG head to PGE.

Completed - Date: _____, by _____

- 4.3.2.3 Transient Pulse Monitor (TPM)
A complete acceptance test was successfully performed by SRI personnel when the TPM was delivered to JPL on 6-25-90. The E-field sensor test procedure given below was included in the ATS10 capture file when the second sun sensor test was run.
- 4.3.2.3.1 Place E-field stimulator over E-field antenna and stimulate E-field.
- 4.3.2.3.2 Run abbreviated test sequence per sections 4.1.2.2 and 4.1.2.3 above except name capture file ATS13.
- 4.3.2.3.3 Compare data from 4.3.2.3.2 with data obtained in step 4.3.2.2.5 above to see changes in event register.
- 4.3.2.3.4 Compare pulse characterization data obtained in step 4.3.2.3.2 with acceptance test or subsystem test data.
- 4.3.2.4 Langmuir Probe (LP)
- 4.3.2.4.1 Connect Langmuir probe head simulator from Air Force to Langmuir probe electronics (LPE).
- 4.3.2.4.2 Run abbreviated test sequence per sections 4.1.2.2 and 4.1.2.3 above except name capture file ATS14.
- 4.3.2.4.3 Compare data from above with acceptance test or subsystem test data.

Data

Raw and Converted Data Readings

<u>1-10</u>		<u>11-20</u>		<u>21-30</u>		<u>31-41</u>	
01	1	F8	248	BD	189	BE	190
1A	26	DF	223	BE	190	BE	190
32	50	C6	198	BE	190	BD	189
4B	75	AD	173	BD	189	BD	189
64	100	94	148	BD	189	BE	190
7D	125	7B	123	BE	190	BD	189
96	150	63	99	BE	190	BD	189
AF	175	4A	74	BD	189	BD	189
C7	199	31	49	BD	189	BD	189
E0	224	18	24	BD	189	BD	189

Data Analysis

The forty-one data points for the sweep voltage define the ramp from -3 V to +5 V then back to -3 V and then the steady +3 V period and finally 0 V. Point 1 then represents -3 V and point 11 represents about +5 V. If point 11 is actually the peak voltage then a linear scale would place the +3 V measurement at 186 which is close to the actual data. The 0 V measurement would be at 96 if the linear scale is used and this is very close to the actual data. The time spent on each part of the sweep voltage function was consistent with the specifications. There is no leakage current data since the furnished probe head simulator did not have detailed instructions. Data was received on the Langmuir probe temperature calculation (89 HEX) but could not be interpreted.

Completed - Date: _____, by _____

- 4.3.2.5 Emitter (Emitter was on for all abbreviated test sequences so this test was not done)
 - 4.3.2.5.1 Connect emitter simulator in GSE with ASIS electronics.
 - 4.3.2.5.2 Run abbreviated test sequence per sections 4.1.2.2 and 4.1.2.3 above except name capture file ATS15.
 - 4.3.2.5.3 Compare data from above with subsystem test data.
- 4.3.2.6 Pressure Sensors
 - Was not requested, however data was reported which showed the pressure sensor channels were working consistently.
- 4.3.3 Housekeeping Tests
 - 4.3.3.1 Power Supply Voltages
 - Was not performed.
 - 4.3.3.2 Temperature
 - Was not requested, however prior bench calibration tests showed a sensitivity of $-0.0279 \text{ V/}^{\circ}\text{C}$ with 20°C being equal to 2.851 V. Temperature changes typically seen in the test data were about 1 or 2 points (e.g., 86 to 85 HEX) during a 3.7 minute test run. This is equivalent to $5 \text{ V}/255$ or 0.0196 V - approximately 0.7°C temperature rise for each point change and a probable start temperature of 28°C .
- 4.3.4 DACS Command Tests
 - 4.3.4.1 Response
 - 4.3.4.1.1 Run through entire set of DACS system commands by selecting the Operations (O) menu and then the Dacs commands (D) menu and then each individual DACS command.

- 4.3.4.1.2 Compare actual PASP Plus system response to commands with expected response.

Data

Each DACS command was exercised and the resultant response was identical to the expected response.

Completed - Date: _____, by _____

4.3.4.2 Sequence

- 4.3.4.2.1 Run system through a complete six orbit data sequence from flight PROM chips by selecting the Test commands (T) menu from the Operations (O) menu and then the Normal test sequence (N). Capture data in a file named OTS5.
- 4.3.4.2.2 Compare actual PASP Plus system sequence with expected sequence.

Data

Each DACS sequence command was exercised and the resultant sequence was identical to the expected sequence.

Completed - Date: _____, by _____

- 4.3.4.3 TPM Commands (Not done. See 4.3.2.3 above)
Select the Send tpm commands (S) menu from the Operations (O) menu and then send the TPM commands 0 through 7. Rename the capture file OTS6.CAP.
- 4.3.4.4 Buffering (Not done. See 4.3.2.3 above)
Use data from 4.3.4.2.1 above to check on buffering of TPM data and inclusion in DACS/spacecraft data stream.
- 4.3.4.5 Telemetry (Not done. See 4.3.2.3 above)
Use data from 4.3.4.2.1 above to check formatting of DACS telemetry data versus specified data format for the real time (RDO) output channels.
- 4.3.5 Integrated System Test
- 4.3.5.1 Input Power
- 4.3.5.1.1 Measure and record in-rush current when system is turned on.
- 4.3.5.1.2 Measure and record noise on input power line when high voltage power supply is operating.
- 4.3.5.1.3 Compare data from tests with system requirements.

Data

In-rush current: 2.5A pk in a 10 msec pulse.
See Photo #9. H-scale is 50 msec./div., V-scale is 500 mA/div.

Reference: Polaroid pictures taken 7-9-90.

Input noise:

11mA pk-pk (4mA RMS) at power lead to HVD board when high voltage supply is on. See Photo #10. H-scale is 2 msec./div., V-scale is 5 mA/div.

380mA pk-pk (134mA RMS) on HVD return lead when high voltage supply is on. See Photo #11. H-scale is 5 usec./div., V-scale is 200 mA/div.

300mV pk-pk (106mA RMS) at power lead to APS when ASIS subsystem is on. See photo #12. H-scale is 5 usec./div., V-scale is 100 mA/div.

Reference: Polaroid photos taken 7-9-90.

Data Analysis

There is no way to determine the effects of these in-rush current and noise levels on the carrier power supply since there is no carrier. However, the values seem reasonable.

Completed - Date: _____, by _____

4.3.5.2 Input Voltage

- 4.3.5.2.1 Make sure that the EPROMs are loaded on the SEQ board of the DACS.
- 4.3.5.2.2 Make sure input voltage is at 28 Volts DC.
- 4.3.5.2.3 Run an abbreviated test sequence and name the capture file ATS16.
- 4.3.5.2.4 Change input voltage to 25 VDC.
- 4.3.5.2.5 Run an abbreviated test sequence and name the capture file ATS17.
- 4.3.5.2.6 Change input voltage to 32 VDC.
- 4.3.5.2.7 Run an abbreviated test sequence and name the capture file ATS18.
- 4.3.5.2.8 Compare data from all three tests with system requirements.

Voltage Data

The voltages from each of the power supplies was checked over the 25-32 VDC input range. Variations in output voltages were not seen to four significant figures. Because of this lack of change the individual test sequences did not show any significant changes.

Current Data

Operating Condition	Input Voltage		
	25	28	32
Low power	0.2A	0.2A	0.2A
W/o HV & Emit	2.3A	2.0A	1.8A
Full power	5.2A	4.9A	4.7A

Data Analysis

The ability of the PASP Plus power supplies to handle the specified range of input voltages was demonstrated. However, the power levels are not consistent with subsequent data in section 4.3.5.3.3 below. Following this test a broken lead was discovered in the HVO Output Cable. This break caused the HVO to remain at about 600 V when the HV was commanded on. This condition created the unusual power consumption readings above.

Completed - Date: _____, by _____

- 4.3.5.3 Power Profile (Additional test)
- 4.3.5.3.1 Measure ASIS input power when IV curves are being measured.
- 4.3.5.3.2 Measure ASIS input power when HV bias voltage is being applied at 500 V with 50K ohm and 5K ohm applied loads. These measurements should be taken with the emitter both off and on.
- 4.3.5.3.3 Measure DACS input power in the low power, standby, and operating modes. During operating mode the TPM, LP, and SS instruments should be operating.

Data

GSE Power Profile at 28V DC.
 Low power mode: 0.2A
 DACS on mode: 1.7A
 HV on (0->500V): 1.8A-2.3A
 Emitter & HV on (0->500V): 3.0A-3.5A

Completed - Date: _____, by _____

- 4.3.5.4 System Thermal Test
Was not performed.

5 Additional Tests

5.1 Array Sunlight Test

- 5.1.1 The PASP Plus system was moved from the laboratory to an outdoors, sunlit area. Two tests were run. In the first test the actual test solar arrays were wired to the GSE electronics and an abbreviated test sequence was run only

for the IV data. Results of this test are shown below as set #1 data and were captured in a file called STS1.CAP. The second test was similar to the first except the demo cart electronics were used, the data is set #2, and the capture file was STS2.CAP.

Data

Array IV Curves - Set #1																
Pt	IV1				IV2				IV3				IV4			
#	V		I		V		I		V		I		V		I	
1	0A	10	AB	171	07	7	6E	110	06	6	7C	124	08	8	BB	187
2	12	18	A7	167	0F	15	6F	111	0B	11	7C	124	0D	13	B8	184
3	1C	28	A6	166	16	22	6E	110	11	17	7C	124	13	19	B7	183
4	25	37	A5	165	1D	29	6F	110	16	22	7C	124	19	25	B7	183
5	2E	46	A2	162	25	37	6E	110	1B	27	7B	123	1F	31	B6	182
6	35	53	9E	158	2D	45	6E	110	20	32	7A	122	24	36	B4	180
7	3D	61	9B	155	34	52	6D	109	26	38	79	121	29	41	B0	176
8	47	71	9A	154	3B	59	6D	109	2C	44	79	121	2F	47	AB	171
9	51	81	9A	154	44	68	6D	109	30	48	79	121	32	50	A9	169
10	5A	90	99	153	4C	76	6C	108	35	53	78	120	3A	58	A8	168
11	64	100	98	152	53	83	6B	107	3C	60	78	120	3F	63	A8	168
12	69	105	99	153	5A	90	69	105	41	65	77	119	44	68	A7	167
13	78	120	98	152	63	99	68	104	49	73	77	119	4A	74	A4	164
14	81	129	96	150	69	105	68	104	4D	77	77	119	52	82	A4	164
15	88	136	93	147	6F	111	67	103	56	86	77	119	57	87	A3	163
16	8C	140	8C	140	77	119	66	102	59	89	77	119	5E	94	A0	160
17	90	144	82	130	7F	127	66	102	5F	95	77	119	62	98	9C	156
18	92	146	78	120	85	133	63	99	66	102	76	118	67	103	98	152
19	95	149	6D	109	9B	139	5F	95	6E	110	76	118	6E	110	96	150
20	97	151	62	98	94	148	5C	92	77	119	75	117	78	120	96	150
21	98	152	56	86	9D	157	57	87	80	128	72	114	81	129	93	147
22	9A	154	49	73	A2	162	50	80	8A	138	6F	111	88	136	8B	139
23	9B	155	3D	61	A6	166	48	72	9B	155	6F	111	97	151	89	137
24	9D	157	30	48	AA	170	3F	63	A8	168	69	105	A2	162	80	128
25	9E	158	22	34	AC	172	36	54	AF	175	5F	95	A9	169	72	114
26	9F	159	15	21	AF	175	2C	44	B5	181	51	81	AE	174	61	97
27	A0	160	07	7	B2	178	21	33	B9	185	43	67	B3	179	4F	79
28	A0	160	07	7	B7	183	00	0	BD	189	33	51	B7	183	3B	59
29	A0	160	07	7	B7	183	00	0	C1	193	23	35	BA	186	27	39
30	A0	160	07	7	B7	183	00	0	C4	196	12	18	BD	189	11	17
31	A0	160	07	7	B7	183	FF	-1	C8	200	01	1	C3	195	FB	-5
32	A0	160	07	7	B7	183	00	0	C8	200	01	1	C3	195	FB	-5

<u>Array IV Curves - Set #2</u>																
Pt	IV1				IV2				IV3				IV4			
#	V		I		V		I		V		I		V		I	
1	09	9	92	146	09	9	85	133	07	7	7F	127	08	8	E1	225
2	12	18	92	146	12	18	85	133	0D	13	7F	127	0F	15	E1	225
3	1A	26	91	145	1A	26	84	132	12	18	7F	127	16	22	E0	224
4	23	35	91	145	22	34	83	131	18	24	7F	127	1C	28	DC	220
5	2D	45	91	145	2B	43	82	130	1E	30	7F	127	22	34	D7	215

6	35	53	91	145	33	51	81	129	23	35	7F	127	28	40	D1	209
7	3E	62	90	144	3C	60	7F	127	2A	42	7F	127	2D	45	CC	204
8	48	72	90	144	42	66	7D	125	30	48	7E	126	35	53	CB	203
9	52	82	90	144	4A	74	79	121	34	52	7E	126	39	57	CB	203
10	5A	90	8F	143	52	82	77	119	3A	58	7C	124	42	66	C9	201
11	64	100	8F	143	59	89	76	118	41	65	7B	123	46	70	C6	198
12	6A	106	8F	143	61	97	75	117	47	71	7B	123	4B	75	C2	194
13	78	120	8C	140	6B	107	73	115	4F	79	7B	123	51	81	BE	190
14	7E	126	89	137	70	112	71	113	53	83	7B	123	5A	90	BD	189
15	84	132	84	132	76	118	70	112	5C	92	79	121	5F	95	BD	189
16	88	136	7C	124	7C	124	6D	109	5F	95	78	120	68	104	BD	189
17	8B	139	74	116	82	130	6B	107	66	102	78	120	6F	111	BD	189
18	8E	142	6A	106	8C	140	6A	106	6F	111	78	120	76	118	BA	186
19	90	144	60	96	95	149	67	103	76	118	78	120	7D	125	B6	182
20	92	146	54	84	9A	154	61	97	7D	125	74	116	88	136	B6	182
21	94	148	49	73	A0	160	5B	91	83	131	6F	111	93	147	B3	179
22	96	150	3D	61	A4	164	53	83	88	136	67	103	9A	154	A9	169
23	97	151	31	49	A7	167	4A	74	8C	140	5E	94	A0	160	9D	157
24	99	153	24	36	AA	170	41	65	90	144	53	83	A6	166	8D	141
25	9A	154	18	24	AC	172	37	55	93	147	48	72	AB	171	7D	125
26	9B	155	0A	10	AE	174	2D	45	97	151	3D	61	AF	175	6B	107
27	9C	156	FD	-3	B0	176	23	35	99	153	31	49	B2	178	59	89
28	9C	156	FD	-3	B5	181	02	2	9C	156	24	36	B5	181	44	68
29	9C	156	FD	-3	B5	181	02	2	9E	158	17	23	B8	184	30	48
30	9C	156	FD	-3	B5	181	02	2	A1	161	09	9	BB	187	1A	26
31	9C	156	FD	-3	B5	181	02	2	A7	167	FB	-5	C1	193	04	4
32	9C	156	FD	-3	B5	181	02	2	A7	167	FB	-5	C1	193	04	4

Array RTD Values

Set	IV1			IV2			IV3			IV4		
#	HEX	DEC	Temp	HEX	DEC	Temp	HEX	DEC	Temp	HEX	DEC	Temp
1	8501	389	50.3	9001	400	45.8	9E01	414	40.0	8301	387	51.1
2	6F01	367	59.3	6A01	362	61.4	9001	400	45.8	6E01	366	59.7

Notes: Temperatures are in °C and are approximate. HEX values can be read by referring to the data analysis at the end of section 2.8.6 above. The tests were run near noon on a very hot, sunny day with the set #2 test being about 25 minutes after the set #1 test. This accounts for the higher array temperatures in set #2.

Data Analysis

The array IV curves and RTD readings were all as expected. There are a few data points which are shown as negative values. These are simply zero offset errors and are not truly negative values. These results were used to plot the full sunlight set of curves in figures 2.3.1.2.2-1 through 2.3.1.2.2-3 of the final report.

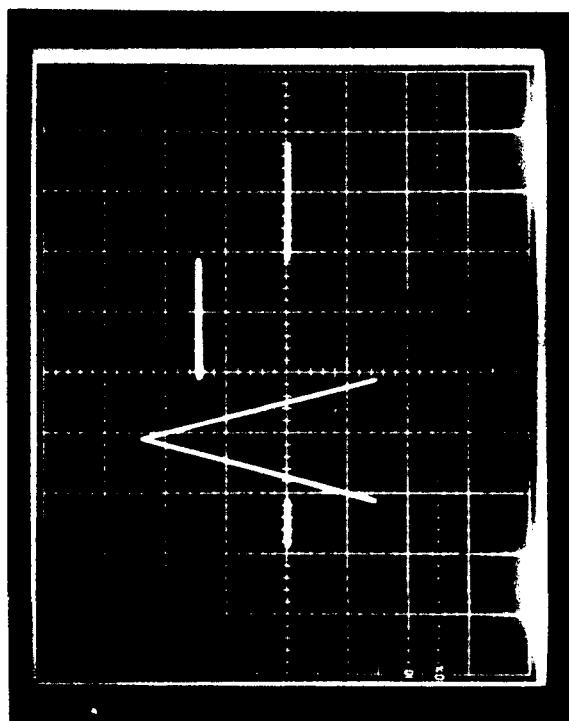


PHOTO #1 — LANGMUIR PROBE SWEEP

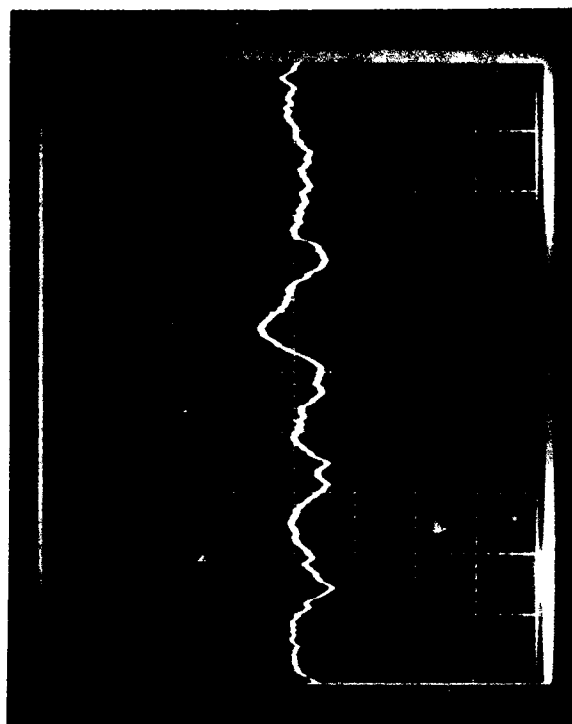


PHOTO #2 — HVPS SET #1, 50V NOISE

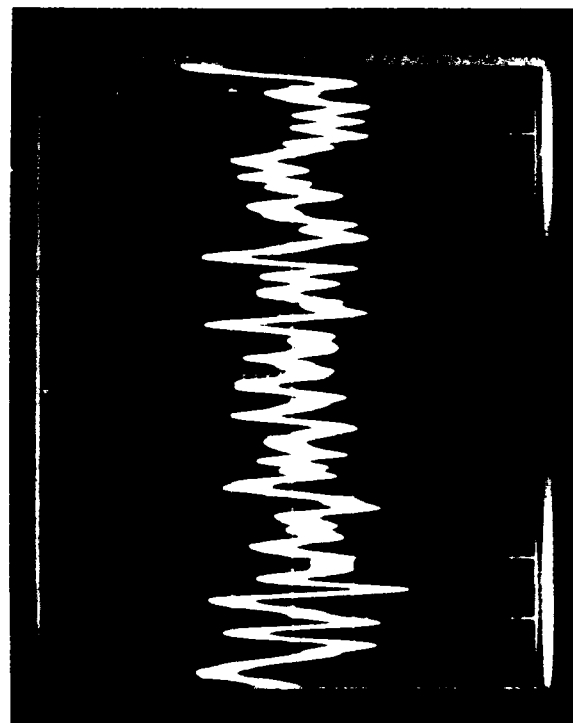


PHOTO #3 — HVPS SET #1, 300V NOISE

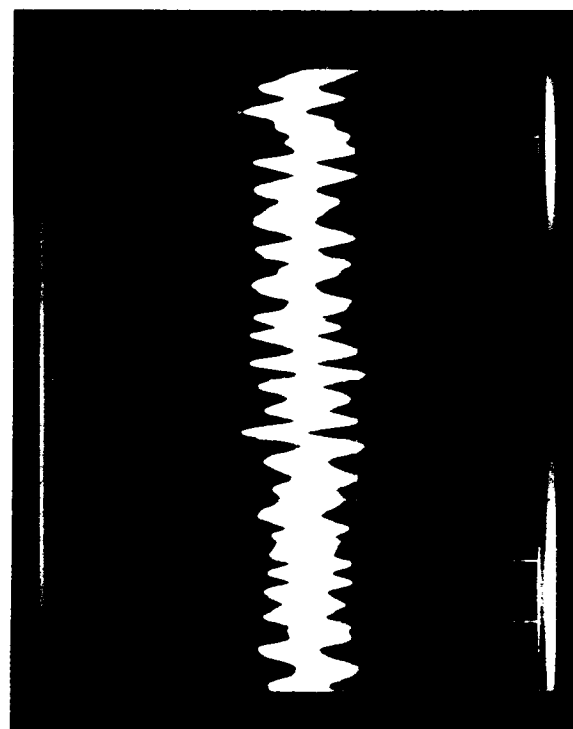


PHOTO #4 — HVPS SET #1, 500V NOISE

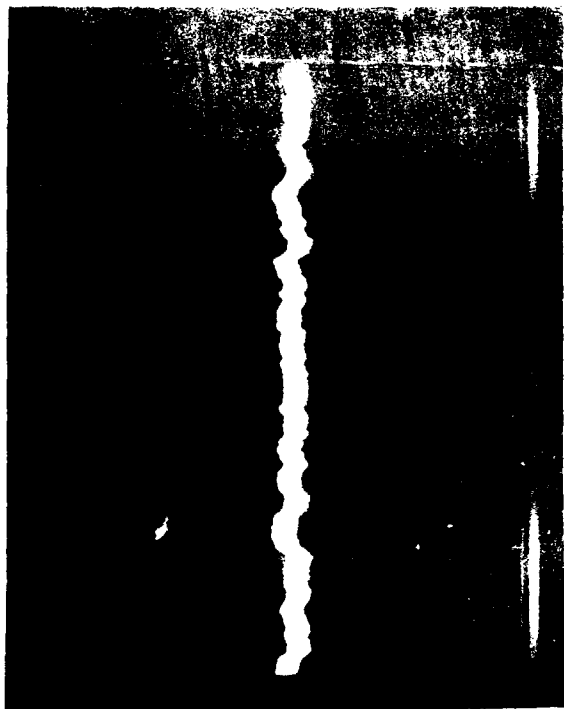


PHOTO #5 — HVPS SET #2, 50V NOISE

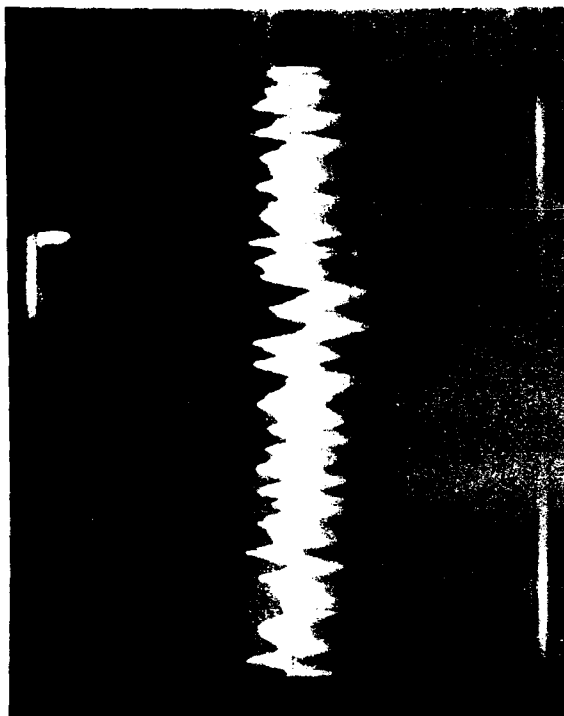


PHOTO #6 — HVPS SET #2, 150V NOISE

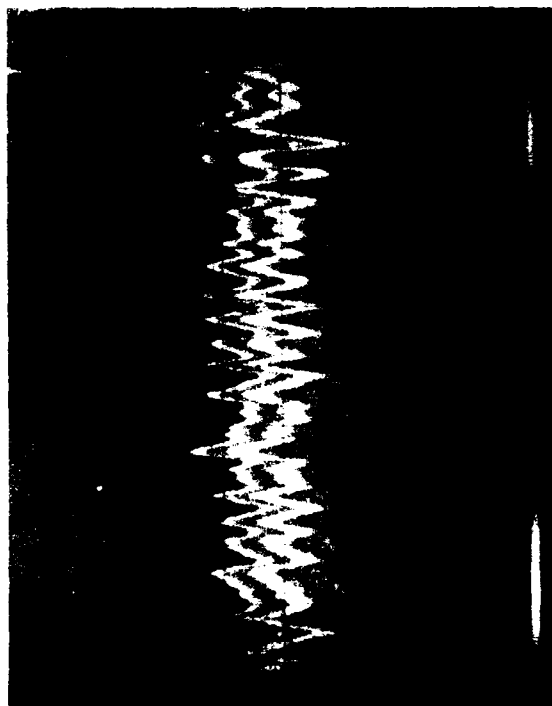


PHOTO #7 — HVPS SET #2, 300V NOISE



PHOTO #8 — HVPS SET #2, 500V NOISE



PHOTO #9 — IN-RUSH CURRENT

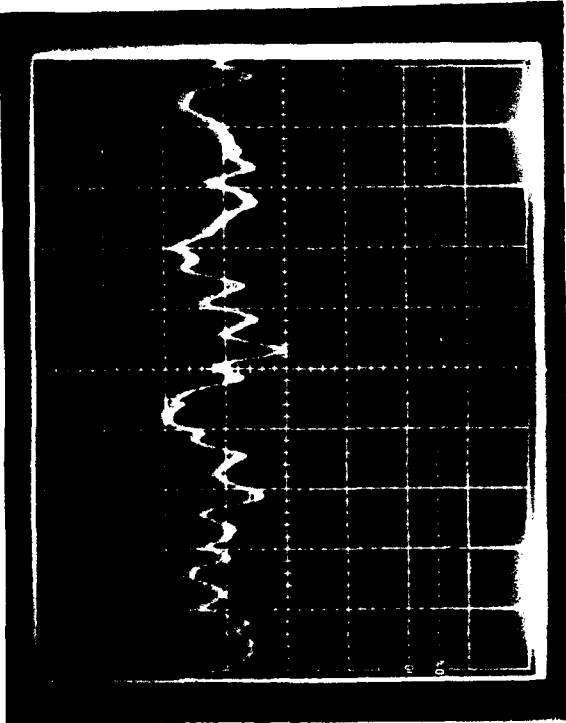


PHOTO #10 — INPUT NOISE

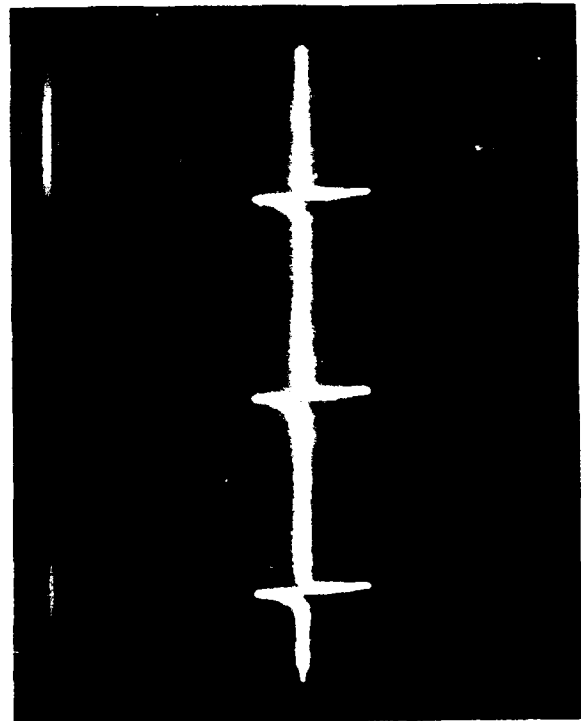


PHOTO #11 — INPUT NOISE

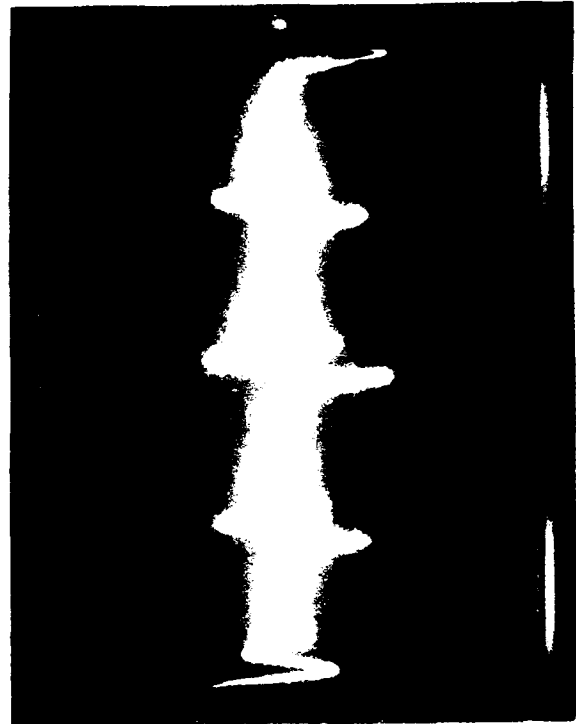


PHOTO #12 — INPUT NOISE

APPENDIX F - PASP Plus ENGINEERING SOFTWARE

1 INTRODUCTION

The development of the PASP Plus experiment required some computer programming for engineering information and documentation. Programs were written to: calculate the parameters of the parallel resistor network for I-V curve generation; do array analysis; plot orbital time-lines; and convert gray code words. Each of these programs are documented here in more detail than can be achieved in the body of the program.

2 GENERAL

All of the programs were written using IBM's version of BASIC. Since most of the programs have screen graphics the Advanced BASIC (BASICA) language was used. This language requires that the computer operator first load the GRAPHICS program into memory before the BASICA program is loaded. If this is done then the graphics outputs that appear on the screen can be captured with a line printer by giving a "Print Screen" command while the graphical output is shown on the screen.

The attributes and options in each program are described in the next section followed by a section which contains program listings.

3 PROGRAM DESCRIPTIONS

3.1 IVA5, I-V Circuitry Calculation

The IVA5 program requires almost all of the available 64K of memory during computation. This program calculates I-V curves from test array module parameters that are stored within the program. Each different array has its own set of parameters. These parameters must be changed if the arrays are changed or a new calibration test indicates new values are needed. First, the array I-V curves at room temperature, low temperature limit and high temperature limit are calculated using the standard single diode formula for solar cell I-V curves. After the I-V curves are calculated the program can be requested to calculate the values of a ladder resistor network. The network resistances at the 32 calculated points will give load values equivalent to the loads applied during the array calibration.

3.1.1 Main Menu - The first menu that is presented allows the choice of any one of the six PASP Plus array modules or a development test module. Once the module is selected by entering the appropriate number, two other choices are presented. First, the low and high temperature limits can be changed or the default values of -100 and 100 degrees C can be accepted. A yes answer accepts the default values. A no answer leads to the entry of new

low and high limit values which can be done using the following format: -NNN,NNN. After the temperature limits choice is made, a choice is presented for printer hard copy or no printer hard copy. After the printer hard copy choice is made the program shifts to the first screen which is a graphical display of the results of the I-V curve calculations.

3.1.2 First Screen - A plot screen is drawn and then three I-V curves are generated from calculations made by the program. The first line is the room temperature (28 °C) I-V curve which is followed by the low temperature I-V curve and then the high temperature I-V curve. When the last curve is drawn another menu appears. Figure 3.1-1 shows a DOS GRAPHICS output of the displayed curves. Figures 3.1-2 through 3.1-5 show pen plotter results from the four arrays used on the PASP Plus experiment.

3.1.3 Second Menu - The second menu contains nine menu choices:

3.1.3.1 Want one load line - This selection draws the load line for any resistance value entered after the "Input Ohms" request. Each load line is retained and displayed with any subsequently requested load lines. Resistance values of each load line are not displayed. This selection is an excellent means to visualize the I-V curve interactions.

3.1.3.2 Uniform 256 line scan - This selection draws a number of load lines depending upon the response to the "Input Scan Max R" request. Input value must be 512 or larger.

3.1.3.3 Staggered line scan - The utility of this choice is doubtful. It allows the selection of three types of scan: 4x64, 8x32, or 16x16. After the scan mode is selected by entering 4, 8 or 16 the "Input Initial Delta R" request is made. Any inputs other than 0 or 1 are acceptable and a number of load lines are displayed.

3.1.3.4 Compute ladder - This selection is most valuable since it calculates 32 resistance value points on the room temperature I-V curve which closely match similar points on the calculated I-V curve. The program requests entry of a binary resistance value approximately twice as large as the target lowest resistance needed for the calculations. Since the resistance ladder is a parallel array of resistances which increase in value by a factor of two, the target lowest resistance value can then be achieved by paralleling all of the resistors. Various combinations of resistors are used to achieve all necessary values between the lowest resistance (nearly short circuit) and an open circuit condition.

3.1.3.5 Print screen - Prints the results displayed on the

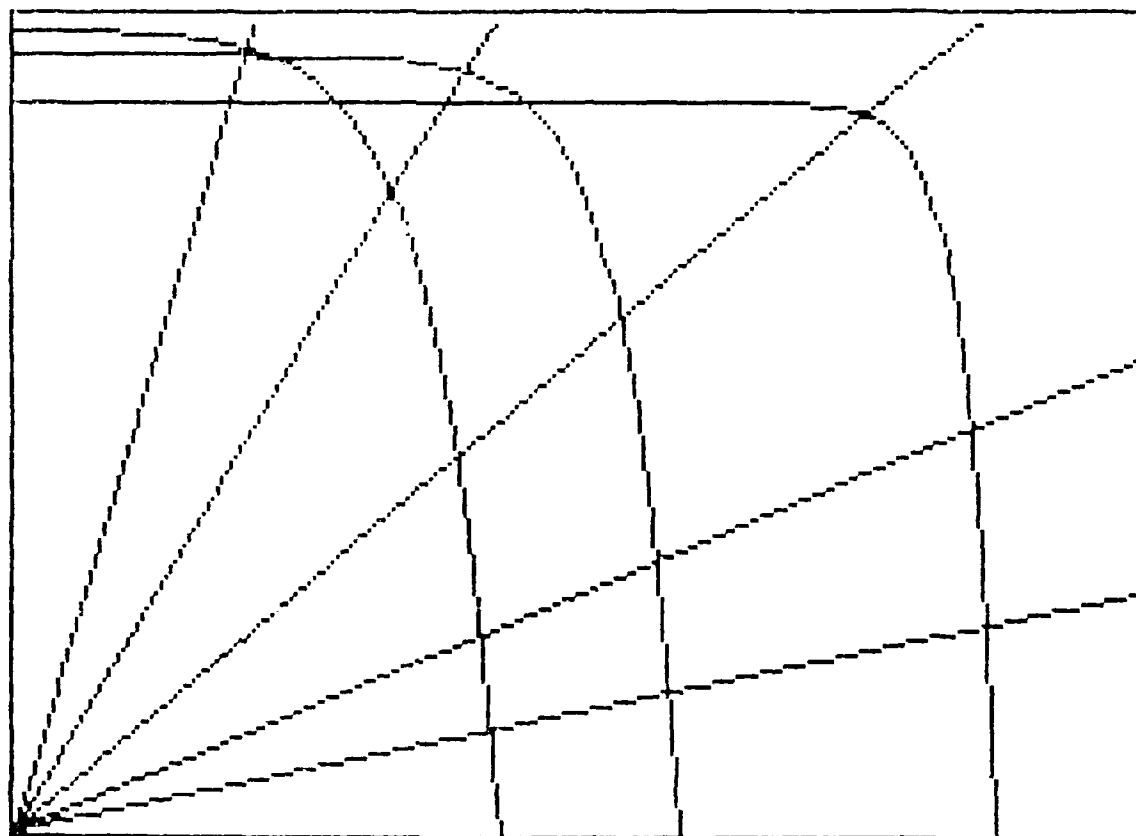
MODULE # 1

TEMPERATURE = 28 DEGREES C
ISC = .322 AMPS
VOC = 59.3 VOLTS
PMAX = 13.85043 WATTS

TEMPERATURE = -100 DEGREES C
ISC = .301392 AMPS
VOC = 87.46 VOLTS
PMAX = 22.626 WATTS

TEMPERATURE = 100 DEGREES C
ISC = .333592 AMPS
VOC = 43.46 VOLTS
PMAX = 8.994538 WATTS

MOD
SP 1
I-FS
.34
V-FS
100
AMPS



VOLTS

Figure 3.1-1 Calculated IV Curves (DOS Graphics)

SILICON PLANAR ARRAY

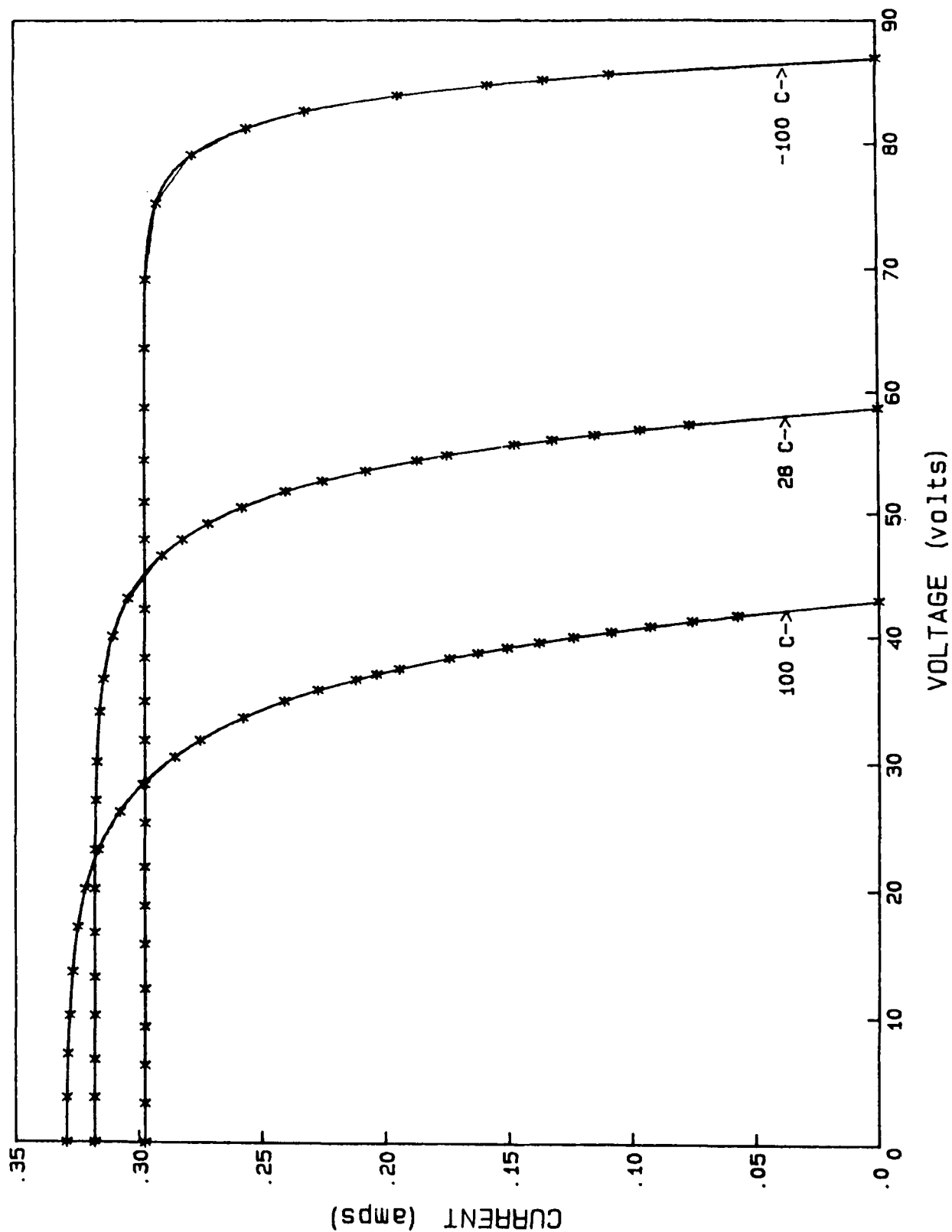


Figure 3.1-2

GALLIUM ARSENIDE PLANAR ARRAY

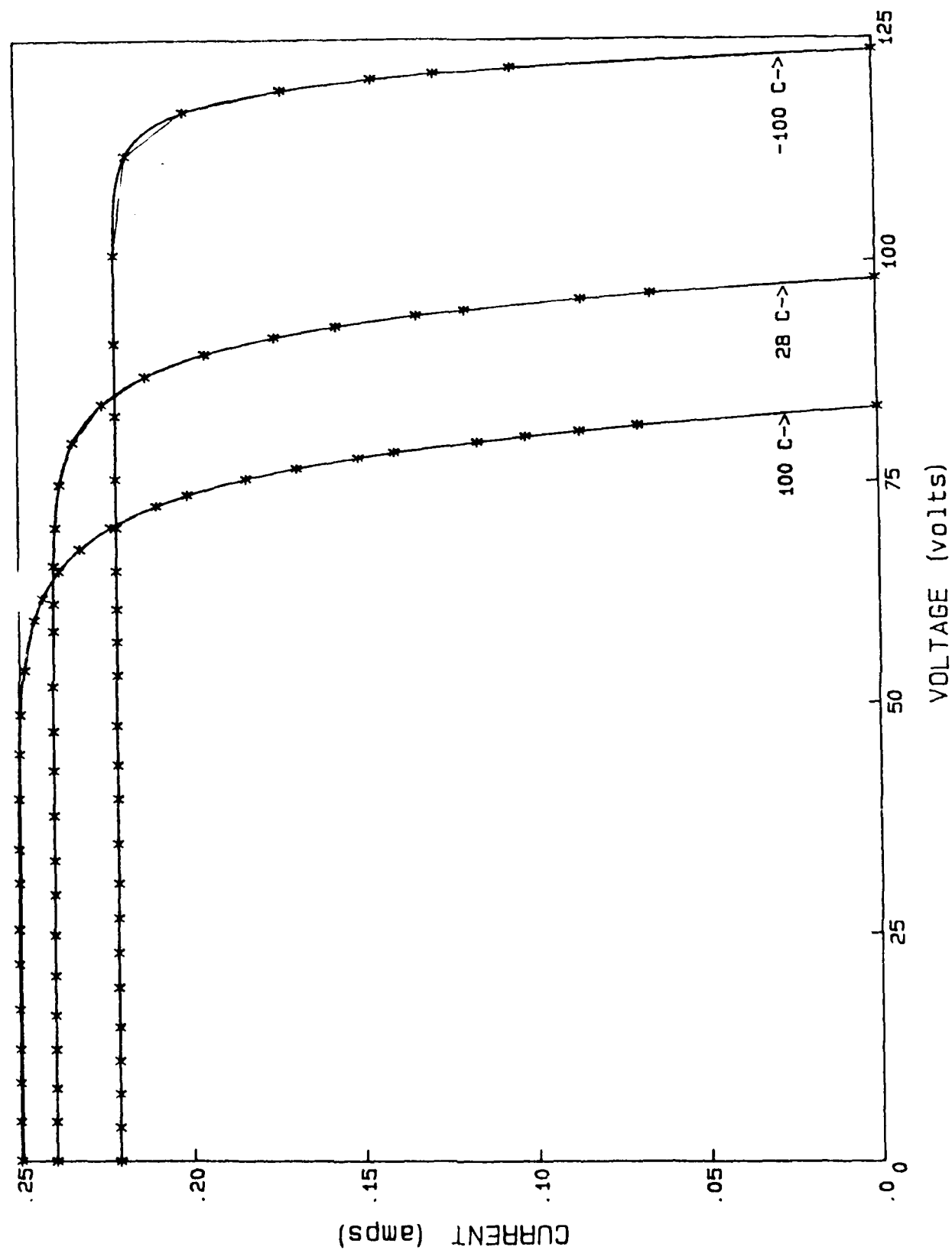


Figure 3.1-3

CASSEGRAINIAN CONCENTRATOR ARRAY

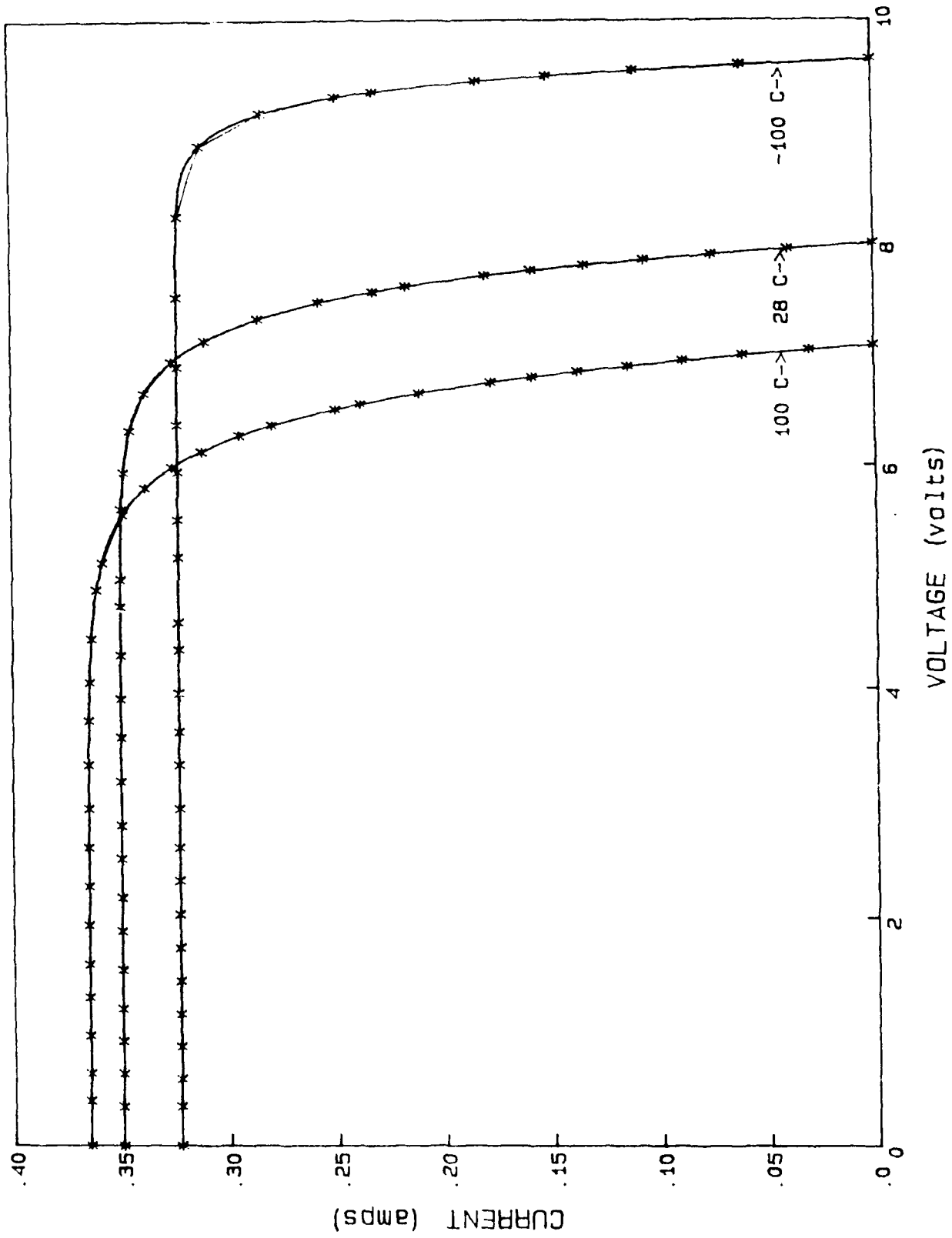


Figure 3.1-4

SLATS CONCENTRATOR ARRAY

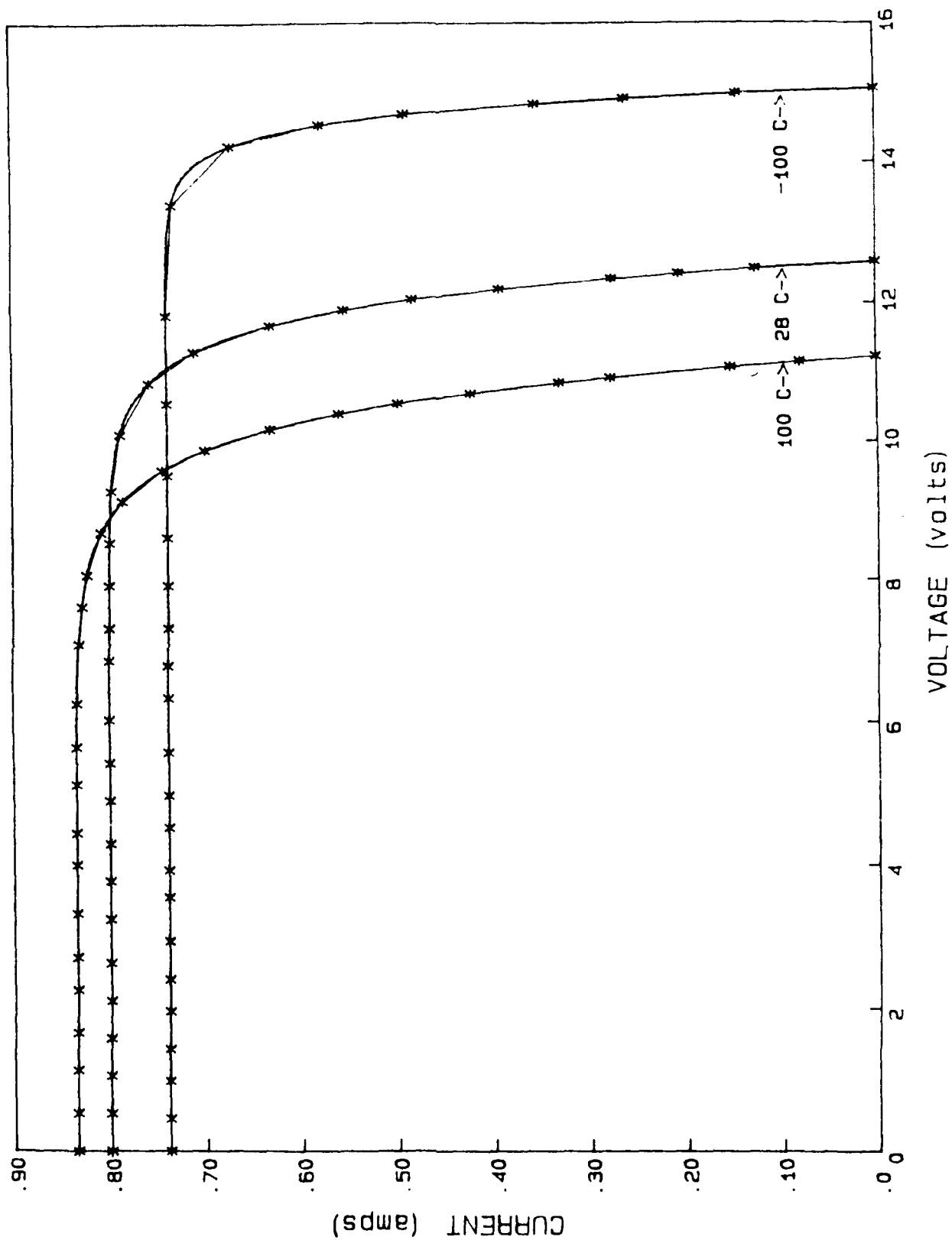


Figure 3.1-5

monitor screen using the printer graphics capability.

3.1.3.6 Return to menu - Returns to the main menu.

3.1.3.7 Overdraw IV with data - This selection should not be requested until after option 4 has been performed. The resultant output is the room temperature I-V calculated curve displayed in white which is then overdrawn with the resistor ladder calculated values displayed in purple.

3.1.3.8 Do penplot - This selection provides an HP7475A pen plotter output of the three I-V curves and the ladder data points rather than the less precise printer graphics output with load lines. This option is also best invoked after the ladder has been computed (option 4). Refer to Figures 3.1-2 through 3.1-5.

3.1.3.9 Quit - Ends the program.

3.2 ARRANAL, Concentrator Array Analysis

The ARRANAL program is used in analyzing, displaying, and reporting on the data taken during acceptance testing of the PASP Plus concentrator array modules. The program requests raw data inputs, displays input data for review, calculates AMO corrections to data, and allows for disk filing and tabular and plotted data reporting.

3.2.1 Main Menu - The main menu lists 22 options:

3.2.1.1 The first eight options, A to H, allow for data entry of both tilt and rotation axis data for both the Mini-Cassegrainian (CC) and the SLATS (SC) concentrator arrays. Selecting a data entry menu option results in being prompted for the following data at given angles: short-circuit current, open-circuit voltage, current and voltage at peak power, and reference cell voltage.

3.2.1.2 The next eight options, I to P, allow for display of raw data and data that has been corrected to AMO values. Use of these options then, allows for review of (not correction of) entered data and subsequent corrected data.

3.2.1.3 The next option, Q, performs the AMO corrections to the raw data.

3.2.1.4 Option R will read data stored on the hard disk. This data must reside on the default drive subdirectory, usually the BASIC subdirectory of the hard disk.

3.2.1.5 Option S saves data to the disk drive - again the BASIC subdirectory of the hard disk.

3.2.1.6 Option T displays the results of an analysis of the data

around the normal to the concentrator base. First, an average of the positive and negative tilt and rotation readings and the final reading is calculated. Values of the high and low extremes of the above data values are then determined. Finally the average, high and low values are displayed along with the percentage deviation of the high and low values around the average.

3.2.1.7 Option U shifts to a menu with six output options:

3.2.1.7.1 Options A and B plot the tilt (see Figure 3.2-1) and rotation (see Figure 3.2-2) axes short circuit current and maximum power data for the CC array while options C and D do the same for the SC array. If hard copy output is desired, the "Shift" and "Print Screen" keys must be pressed simultaneously within four seconds of the end of the screen plot activity. At the end of the graphics output press any key to continue.

3.2.1.7.2 Option E is used to print out tabular reports on the raw and corrected data for both axes and arrays. For report output the instructions are to activate the printer by pressing "Ctrl" and "Print Screen" simultaneously. At the end of the report sequence the printer should be deactivated by again pressing these two keys.

3.2.1.7.3 Option F returns to the main menu.

3.2.1.8 Option V ends the program.

3.3 ORBITPP, Plot of Orbital Data Collection Periods

The ORBITPP program produces an HP7475A penplotter output of the various data collection activities by the PASP Plus instrument in a typical 90 minute noon to midnight, polar shuttle orbit. Sunlight time is assumed to be 53 minutes while occultation time is assumed to be 37 minutes. PASP Plus data collection is split into three data acquisition elements of about 17 minutes each so that the arctic, equatorial and austral regions can be covered. Each of the data acquisition elements is further divided into five environmental data periods (Langmuir Probe and array I-V curves) which are interspersed with four, longer high voltage bias periods. The most important time driver is the dwell time at each high voltage level. There are eleven high voltage levels - zero, five positive, zero and then four negative.

3.3.1 Main Menu - The main menu is really a series of prompted data entries:

3.3.1.1 Num of pos voltage steps - Five is the number selected for the present PASP Plus sequence, however since no provision has been made in this program for a zero voltage dwell, six steps should be entered.

3.3.1.2 Pos dwell (secs) - Twenty seconds has been selected for

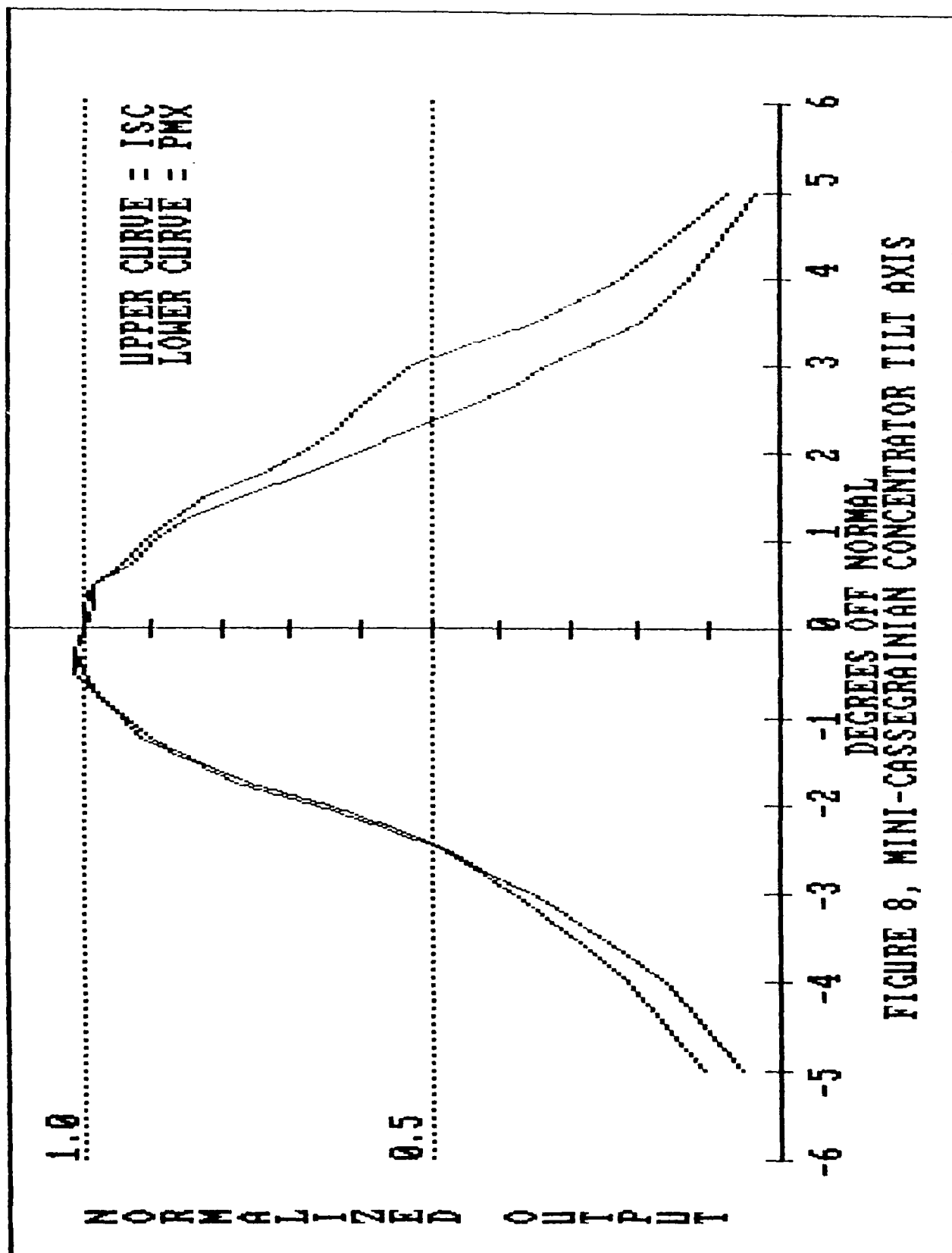


Figure 3.2-1

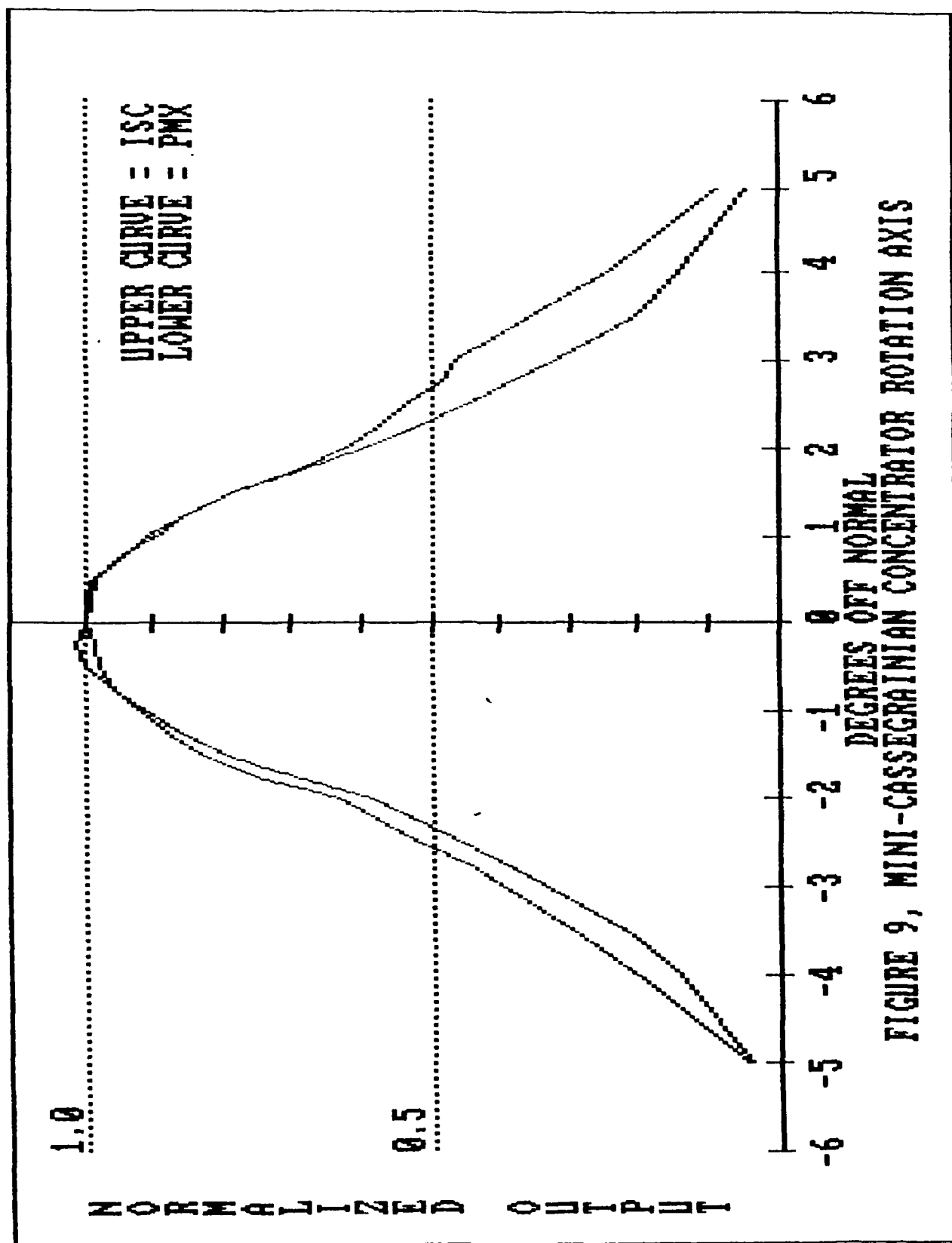


Figure 3.2-2

PASP Plus.

3.3.1.3 Num of neg voltage steps - Five is the number selected for the present PASP Plus sequence (includes zero start state).

3.3.1.4 Neg dwell (secs) - Twenty seconds has been selected for PASP Plus.

3.3.1.5 Num of scans/orbit - Three scans will be made.

3.3.1.6 Num of modules tested/scan - Only four modules can be tested during each scan period. This fact is based primarily on the decision to dwell for 20 seconds at each high voltage scan point.

3.3.1.7 Sun acquisition time (secs) - This figure is not a hard figure since it is spacecraft hardware dependent. Sun acquisition time must include time required for orientation. The figure usually used is 60 seconds.

3.3.1.8 LP and IV scan time (secs) - The LP scan time is 2 seconds and the IV scan time for all six modules is around 6 seconds. A total time of 10 seconds is usually used.

3.3.2 Pen Plotting - At the end of the prompted screen data entries the program immediately starts into its pen plotting mode. The plotter should be ready for this output. The program assumes that the plotter is on serial port #1 (COM1) and set for 2400 baud, space parity, seven data bits, and one stop bit. See Figure 3.3-1 for typical output.

3.4 GRAY, Gray Code Calculations for Adcole Sun Sensor

The GRAY program is used to calculate the various permutations of the Adcole sun sensor gray code. A gray code is used to enhance data reliability.

3.4.1 Main Menu - The main menu contains seven options:

3.4.1.1 Input gray, compute binary and decimal - An eight bit gray word is entered. This option allow the calculation of binary and decimal values to check on data transmission accuracy.

3.4.1.2 Input decimal, compute binary and gray - Similar to option 1 above with maximum decimal value of 255.

3.4.1.3 Input decimal top, compute binary and gray - This option requests entry of the top decimal number in a range (255 maximum). The program then calculates binary and gray scale values from zero to the top number specified. Decimal, binary and gray values are displayed on the screen.

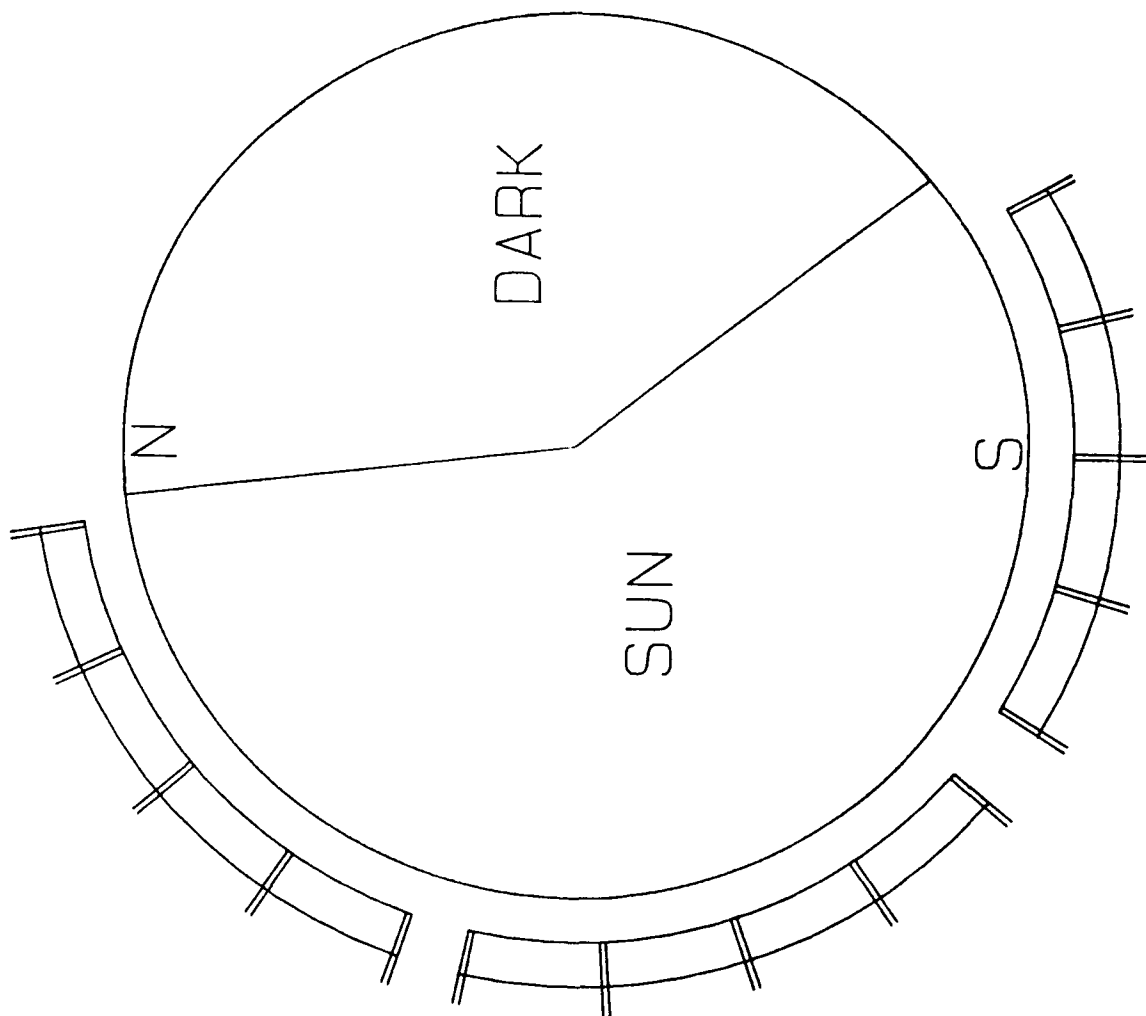


Figure 3.3-1 Orbit Pen Plot

3.4.1.4 Input 16 bit gray word, compute binary and decimal parts - The 16 bit gray word consists of 8 bits of X-axis data followed by 8 bits of Y-axis data from the sun sensor. The binary and decimal values for the X- and Y-axes are displayed along with the associated angles. The sun pointing angle is also displayed.

3.4.1.5 Input pointing angles, compute gray word - Entering pointing angles results in display of the entered angles along with the sun pointing angle and the associated 16 bit gray word.

3.4.1.6 Print a table of values, gray code and angles - The tabular values are decimals from 0 to 255 along with the associated eight bit binary and its hex, the eight bit gray and its hex, and the angle which ranges from -63.759° to $+63.759^\circ$. See Table 3.4-1 (attached) for output values.

3.4.1.7 Quit - Ends program.

4 PROGRAM LISTINGS

4.1 IVA5.BAS Listing

```

10 'IVA5, PASP IV CIRCUIT CALCULATIONS PROGRAM BY G. W. DOWNING
20 'THIS PROGRAM CALCULATES PARALLEL RESISTOR LADDER VALUES AND RESULTING IV
30 'CURVE SHAPES FOR ARRAY PARAMETERS CONTAINED IN SUBROUTINES 1280 THRU 1480
40 '***** USE ONLY UPPER CASE LETTERS FOR ENTRIES *****
50 '****** LOAD "GRAPHICS" PROGRAM BEFORE RUNNING "BASICA" PROGRAM *****
60 DIM SS(9604),RT(255):AS="AMPS":BS="VOLTS":CS=" F.S.":KT=.05
70 DIM VC(2,200),IC(2,200),VD(2,200),ID(2,200)
80 DIM IS(4),VO(4),PM(4),KT(4)
90 TS(1)="-100 C->":TS(0)=" 28 C->":TS(2)=" 100 C->"
100 LPRINT CHR$(27)"I"CHR$(5)
110 GOSUB 1020
120 GOSUB 1660
130 VX=VO-VT*(28-LT)
140 C=1:T(2)=28:GOSUB 620
150 T(3)=LT:GOSUB 760
160 GOSUB 620:KT=KT(1):IS=IS(1):VO=VO(1)
170 T(4)=HT:GOSUB 860
180 GOSUB 620:KT=KT(1):IS=IS(1):VO=VO(1)
190 IF Z$="Y" THEN GOSUB 940
200 GET (0,0)-(WX,WY-WY/100),SS
210 REM-----DRAW LOADLINES-----
220 LOCATE 23,1:PRINT"WANT ONE LOAD LINE (1), UNIFORM 256"
230 PRINT"LINE SCAN (2), STAGGERED LINE SCAN (3),"
240 PRINT"COMPUTE LADDER (4), PRINT SCREEN (5),"
250 PRINT"RETURN TO MENU (6), OVERDRAW IV WITH"
260 INPUT"DATA (7), DO PENPLOT (8), OR QUIT (9) ";LL
270 IF LL<1 OR LL>9 THEN 260
280 IF LL=2 THEN 400
290 IF LL=3 THEN 460
300 IF LL=4 THEN 1790
310 IF LL=5 THEN 1520
320 IF LL=6 THEN 1640
330 IF LL=7 THEN 2400 'Run option 4 first
340 IF LL=8 THEN 2550
350 IF LL=9 THEN END
360 LOCATE 24,1:INPUT"INPUT OHMS (0 RETURNS TO MENU) ";OM:IF OM=0 THEN 220
370 GOSUB 1660:PUT (0,0),SS,PSET:GOSUB 910
380 GET (0,0)-(WX,WY-WY/100),SS
390 GOTO 360
400 REM-----UNIFORM 256-----
410 LOCATE 24,1:INPUT"INPUT SCAN MAX R ";OT
420 GOSUB 1660:PUT (0,0),SS,PSET:DE=OT/256

```

```

430 FOR J=0 TO OT STEP DE:OM=J:IF OM=0 THEN 450
440 GOSUB 910
450 NEXT:GOTO 210
460 REM-----STAGGERED LOADLINE, VALUE 4-----
470 LOCATE 24,1:INPUT"4X64, 8X32, OR 16X16 SCAN (4/8/16) ";SC
480 IF SC<>4 AND SC<>8 AND SC<>16 THEN 470
490 LOCATE 24,1:INPUT"INPUT INITIAL DELTA R ";DZ
500 GOSUB 1660:PUT (0,0),SS,PSET
510 IF SC=16 THEN 820
520 IF SC=8 THEN 580
530 OM=0:CA=0:FOR J=0 TO 3:S=2^J
540 FOR K=1 TO 64 STEP S:OM=OM+DZ*(2^J)
550 CA=CA+1:GOSUB 910:NEXT K:NEXT J
560 GOTO 210
570 REM-----STAGGERED LOADLINE, VALUE 8-----
580 OM=0:CA=0:FOR J=0 TO 7:S=2^J
590 FOR K=1 TO 32 STEP S:OM=OM+DZ*(2^J)
600 CA=CA+1:GOSUB 910:NEXT K:NEXT J
610 GOTO 210
620 REM-----DRAW I-V CURVE SUBROUTINE-----
630 PM=0:D=0:PSET(0,IS):CC=0
640 IO=IS/(2.71828^(VO/(NC*KT)))
650 FOR V=VX/200 TO VX*1.005 STEP (VX/200)
660 D=D+1:E=2.71828^((V/NC)/(KT))-1
670 I=IS-(IO*E)
680 P=I*V:IF P>PM THEN PM=P
690 CC=CC+1:IF CC=200 THEN 710
700 VC(C-1,CC)=V:IC(C-1,CC)=I
710 LINE -(V,I):IF V>VO OR CC=200 THEN 730
720 NEXT
730 IC(C-1,0)=IS:VC(C-1,CC)=VO:IC(C-1,CC)=0:ID(C-1,0)=IS:VD(C-1,50)=VO
740 C=C+1:IS(C)=IS:VO(C)=VO:PM(C)=PM
750 RETURN
760 REM-----COMPUTE LOW TEMP CORRECTIONS-----
770 KT(1)=KT:KT=KT*(273+LT)/300
780 IS(1)=IS:IS=IS*(1-(IT*(28-LT)))
790 VO(1)=VO:VO=VO-VT*(28-LT)
800 RETURN
810 REM-----STAGGERED LOADLINE, VALUE 16-----
820 OM=0:CA=0:FOR J=0 TO 15:S=2^J
830 FOR K=1 TO 16 STEP S:OM=OM+DZ*(2^J)
840 CA=CA+1:GOSUB 910:NEXT K:NEXT J
850 GOTO 210
860 REM-----COMPUTE HIGH TEMP CORRECTIONS-----
870 KT(1)=KT:KT=KT*(273+HT)/300
880 IS(1)=IS:IS=IS*(1+(IT*(HT-28)))
890 VO(1)=VO:VO=VO+VT*(HT-28)
900 RETURN
910 REM-----DO SINGLE LOADLINE-----
920 PSET (0,0):LINE -(WX,WX/OM)
930 RETURN
940 REM-----PRINT SUBROUTINE-----
950 LPRINT"MODULE #"Z
960 FOR J=2 TO 4
970 LPRINT:LPRINT"TEMPERATURE = "T(J)"DEGREES C"
980 LPRINT"ISC ="IS(J)"AMPS"
990 LPRINT"VOC ="VO(J)"VOLTS"
1000 LPRINT"PMAX ="PM(J)"WATTS"
1010 NEXT:RETURN
1020 REM-----MENU-----
1030 SCREEN 1:COLOR 9,1:KEY OFF
1040 CLS:PRINT:PRINT" -MENU-":PRINT
1050 PRINT"1. SP MODULE - SILICON":PRINT"2. GP MODULE - GAAS"
1060 PRINT"3. CC MODULE - GAAS CASS CONC"
1070 PRINT"4. SC MODULE - GAAS SLAT CONC"
1080 PRINT"5. ICA MODULE - SILICON PASS
1090 PRINT"6. TBD"
1100 PRINT"7. TEST MODULE - SILICON
1110 PRINT:PRINT" INPUT CHOICE "
1120 Z$=INPUT$(1):Z=VAL(Z$)
1130 IF Z<1 OR Z>7 THEN 1120
1140 PRINT Z:PRINT:PRINT" ARE MIN/MAX TEMPS -100,100 C (Y/N) "
1150 Z$=INPUT$(1):PRINT" ";Z$

```

```

1160 IF Z$<>"Y" AND Z$<>"N" THEN 1150
1170 IF Z$="Y" THEN LT=-100:HT=100:GOTO 1190
1180 PRINT:INPUT" MIN AND MAX TEMPS (LOW, HIGH IN DEG C) = ";LT,HT
1190 PRINT:PRINT" PRINTER OUTPUT (Y/N)"
1200 Z$=INPUT$(1):PRINT" ";Z$
1210 IF Z$<>"Y" AND Z$<>"N" THEN 1200
1220 IF Z = 1 THEN GOSUB 1280:RETURN
1230 IF Z = 2 THEN GOSUB 1320:RETURN
1240 IF Z = 3 THEN GOSUB 1360:RETURN
1250 IF Z = 4 THEN GOSUB 1400:RETURN
1260 IF Z = 5 THEN GOSUB 1440:RETURN
1270 IF Z = 7 THEN GOSUB 1480:RETURN
1280 REM-----SP MODULE VARIABLES-----
1290 IT=.0005:VT=-.22:IS=.322:VO=59.3:WY=.34:WX=100:NC=100:ZN$="SP"
1300 SX$="910":SY$="750":ZT$="SILICON PLANAR ARRAY"
1310 RETURN
1320 REM-----GP MODULE VARIABLES-----
1330 IT=.000001E-04:VT=-.2:IS=.235:VO=93.7:WY=.26:WX=125:NC=100:ZN$="GP"
1340 SX$="525":SY$="550":ZT$="GALLIUM ARSENIDE PLANAR ARRAY"
1350 RETURN
1360 REM-----CC MODULE VARIABLES-----
1370 IT=.000001E-04:VT=-.0128:IS=.362:VO=9:WY=.4:WX=11:NC=8:ZN$="CC"
1380 SX$="502":SY$="850":ZT$="CASSEGRAINIAN CONCENTRATOR ARRAY"
1390 RETURN
1400 REM-----SC MODULE VARIABLES-----
1410 IT=.000001E-04:VT=-.0192:IS=.52:VO=12:WY=.6:WX=16:NC=12:ZN$="SC"
1420 SX$="802":SY$="9100":ZT$="SLATS CONCENTRATOR ARRAY"
1430 RETURN
1440 REM-----ICA MODULE VARIABLES-----
1450 IT=.0005:VT=-.018:IS=.197:VO=4.36:WY=.22:WX=7:NC=8:ZN$="IC"
1460 SX$="701":SY$="750":ZT$="INTEGRAL COVER PLANAR ARRAY"
1470 RETURN
1480 REM-----TEST MODULE VARIABLES-----
1490 IT=.0005:VT=-9.000001E-02:IS=1:VO=21:WY=1.2:WX=34:NC=36:ZN$="TE"
1500 ZT$="TEST PLANAR ARRAY"
1510 RETURN
1520 REM-----PRINT SCREEN-----
1530 PRINT:PRINT" AFTER SCREEN HAS BEEN REDRAWN, PRESS"
1540 PRINT" SHIFT/PRISC KEY TO PRINT SCREEN."
1550 PRINT" AFTER PRINT SCREEN IS DONE PRESS ANY"
1560 PRINT" KEY TO CONTINUE. PRESS ANY KEY NOW"
1570 PRINT" WHEN READY."
1580 XS=INPUT$(1):IF XS="" THEN 1580
1590 GOSUB 1660:PUT (0,0),SS,PSET
1600 XS=INPUT$(1):IF XS="" THEN 1600
1610 LPRINT CHR$(12)
1620 GOTO 210
1630 REM-----RETURN TO MENU-----
1640 VIEW:CLS:GOTO 110
1650 REM-----PLOT WINDOW-----
1660 VIEW:CLS
1670 VIEW (41,1)-(280,160),,3
1680 WINDOW (0,0)-(WX,WY)
1690 FOR I=1 TO 4
1700 LOCATE 9+I,4:PRINT MID$(AS,I,1):NEXT
1710 LOCATE 22,17:PRINT B$
1720 LOCATE 1,1:PRINT"MOD"
1730 LOCATE 2,1:PRINT ZN$:Z
1740 LOCATE 4,1:PRINT"I-FS"
1750 LOCATE 5,1:PRINT WY
1760 LOCATE 7,1:PRINT"V-FS"
1770 LOCATE 8,1:PRINT WX
1780 RETURN
1790 REM-----SELECT LOADLINE-----
1800 PI=3.141593:RT=WX*TAN(PI/90)/WY
1810 LOCATE 24,1:PRINT"TARGET R =";RT;"OHMS."
1820 PRINT"INPUT LOW R OF LADDER (APPROX 2*TARGET
1830 INPUT"R USING A BINARY MULTIPLE) ";DR
1840 FOR I=0 TO 7:R(I)=DR*2^I:NEXT I
1850 IF Z$="N" THEN 1890
1860 LPRINT:LPRINT"THE VALUES OF THE LADDER RESISTORS IN OHMS ARE:"LPRINT
1870 FOR I=0 TO 3:LPRINT R(I),:NEXT I:LPRINT:FOR I=4 TO 7:LPRINT R(I),:NEXT I
1880 LPRINT

```

```

1890 LB=0:CA=0:DR=DR/2:OM=DR
1900 PRINT:PRINT"I'M WORKING, YOU TAKE A BREAK"
1910 GOSUB 2180
1920 IF Z$="N" THEN 1950
1930 LPRINT:LPRINT"POINT","BYTE","BYTE","BYTE"," R"," R"
1940 LPRINT"NUMBER":"DEC","HEX","BINARY","ACTUAL","TARGET":LPRINT
1950 GOSUB 1660:PUT (0,0),SS,PSET:KK=1
1960 FOR I=2 TO 88 STEP 2:AG=I*PI/180
1970 RT=WX*TAN(AG)/WY:CA=CA+1:AA=100
1980 FOR J=1 TO 255:AP=ABS(RT(J)-RT)
1990 IF AP<AA THEN AA=AP:K=J
2000 NEXT J
2010 KK=K
2020 IF RT(K)=OM THEN CA=CA-1:GOTO 2120
2030 IF Z$="N" THEN 2110
2040 D=K:GOSUB 2260
2050 LB$=HEX$(K):IF LEN(LB$)=1 THEN LB$="0"+LB$
2060 LPRINT USING "###"      ",CA;K;
2070 LPRINT USING "&";LB$;
2080 LPRINT TAB(42);:LPRINT USING "&";D$;
2090 LPRINT TAB(56);:LPRINT USING "###.##";RT(K);
2100 LPRINT TAB(69);:LPRINT USING "###.##";RT
2110 OM=RT(K):GOSUB 910:GOSUB 2310
2120 NEXT I
2130 FOR I=0 TO 2:VD(I,CA+1)=VD(I,50):NEXT I
2140 IF Z$="N" THEN 2170
2150 LPRINT CA+1 TAB(17)"0"TAB(29)"00"TAB(42)"00000000"TAB(57)"INF"TAB(70)"INF"
2160 LPRINT CHR$(12);
2170 GET (0,0)-(WX,WY-WY/200),SS:BEEP:GOTO 210
2180 REM-----BUILD R TABLE-----
2190 FOR I=0 TO 255:D=I:GOSUB 2260
2200 FOR J=0 TO 7:IF MID$(D$,8-J,1)="0" THEN 2230
2210 IF RT(I)=0 THEN RT(I)=R(J):GOTO 2230
2220 RT(I)=RT(I)*R(J)/(RT(I)+R(J))
2230 NEXT J
2240 REM LPRINT I,RT(I)
2250 NEXT I:RETURN
2260 REM-----DECIMAL TO BINARY-----
2270 ZZ=0:D$="":FOR L=7 TO 0 STEP-1
2280 ZZ=D-2^L:IF ZZ<0 THEN D$=D$+"0":GOTO 2300
2290 D=ZZ:D$=D$+"1"
2300 NEXT L:RETURN
2310 REM-----STORE IV DATA PAIR-----
2320 FOR M=0 TO 2
2330 FOR J=1 TO 200:IF (VC(M,J)/IC(M,J))<OM THEN 2370
2340 IF ABS((VC(M,J)/IC(M,J))-OM)<ABS((VC(M,J-1)/IC(M,J-1))-OM) THEN 2360
2350 VD(M,CA)=VC(M,J-1):ID(M,CA)=IC(M,J-1):GOTO 2380
2360 VD(M,CA)=VC(M,J):ID(M,CA)=IC(M,J):GOTO 2380
2370 NEXT J
2380 NEXT M
2390 RETURN
2400 REM-----OVERDRAW 28C IV CURVE WITH DATA POINTS-----
2410 ON Z GOSUB 1280,1320,1360,1400,1440,1480
2420 GOSUB 1660
2430 PM=0:D=0:PSET(0,IS)
2440 IO=IS/(2.71828^(VO/(NC*KT)))
2450 FOR V=VO/100 TO (VO*1.01)STEP (VO/100):D=D+1
2460 E=2.71828^((V/NC)/KT)-1
2470 I=IS-(IO*E)
2480 LINE -(V,I):NEXT
2490 PSET(VD(0,0),ID(0,0))
2500 FOR I=1 TO 30
2510 LINE -(VD(0,I),ID(0,I)),2:NEXT
2520 LOCATE 24,1:PRINT"PRESS ENTER TO CONTINUE"
2530 X$=INPUT$(1):IF X$="" THEN 2530
2540 GOTO 220
2550 '-----SET UP HP PENPLOT-----
2560 '-----SET AREA, SCALE, AND AXES-----
2570 PH=VAL(LEFT$(SX$,1)):SX=VAL(RIGHT$(SX$,LEN(SX$)-1)):FX=PH*SX
2580 PV=VAL(LEFT$(SY$,1)):SY=VAL(RIGHT$(SY$,LEN(SY$)-1)):FY=PV*SY
2590 OPEN "COM1:2400,S,7,1,RS,CS65535,DS,CD" AS #1
2600 PRINT #1,"IN;IP 1500,1160,9670,7285;SC 0,";FX;"0,";FY;"
2610 PRINT #1,"SP1;VS10PUPA0,";FY;"PDPA0,OPA";FX;"OPA";FX;FY;"PA0,";FY;"PUVS;"

```

```

2620 '-----TIC AXES-----
2630 PRINT #1,"TL1.4,0;PUPAO,OPD;"
2640 FOR I=1 TO PH
2650 PRINT#1,"PR";SX;"",0;XT;"
2660 NEXT
2670 PRINT #1,"TL1,0;PUPAO,OPD;"
2680 FOR I=1 TO PV
2690 PRINT#1,"PRO,";SY;"YT;"
2700 NEXT:PRINT#1,"PU;"
2710 '-----ANNOTATE-----
2720 FOR I=0 TO PH:X=I*SX
2730 PRINT#1,"PA";X;"",0CP-1.8,-0.8;LB";X;""+CHR$(3)
2740 NEXT
2750 PRINT#1,"SR1,2PA";.4*FX,-.077*FY;"LBVOLTAGE (volts)" +CHR$(3)
2760 PS=((32-LEN(ZT$)*.5)/64)*FX
2770 PRINT#1,"SR1,2PA";PS,1.03*FY;"LB";ZT$;""+CHR$(3)
2780 FOR I=0 TO PV:Y=I*SY:Y$=STR$(Y):IF LEN(Y$)=3 THEN Y$=" 0"+RIGHT$(Y$,2)
2790 Y$="." +MID$(Y$,2,2)
2800 PRINT#1,"SRPA0,";Y;"CP-3.3,-0.3;LB";Y$;""+CHR$(3)
2810 NEXT
2820 PRINT#1,"PA";-.056*FX,.37*FY;"SR1,2;"
2830 PRINT#1,"DIO,1;LBCURRENT (amps)" +CHR$(3)
2840 PRINT#1,"PUDISR;"
2850 '-----PLOT CALCULATED IV'S-----
2860 FOR I=0 TO 2
2870 IC=IC(I,0)*1000
2880 PRINT#1,"PA";VC(I,0);IC;"PD;"
2890 FOR J=1 TO 200
2900 IC=IC(I,J)*1000
2910 PRINT#1,"PA";VC(I,J);IC;"
2920 IF VC(I,J)=0 AND IC(I,J)=0 THEN 2940
2930 NEXT J
2940 PRINT #1,"PU;"
2950 NEXT I
2960 '-----PLOT IV DATA POINTS-----
2970 FOR I=0 TO 2
2980 ID=ID(I,0)*1000
2990 PRINT#1,"SM";PA";VD(I,0);ID;"PD";":VM=0
3000 FOR J=1 TO 200
3010 ID=ID(I,J)*1000
3020 IF VD(I,J)>VM THEN VM=VD(I,J)
3030 IF VD(I,J)=0 AND ID(I,J)=0 THEN 3060
3040 PRINT#1,"PA";VD(I,J);ID;"
3050 NEXT J
3060 PRINT#1,"PUSH;"
3070 PRINT#1,"PA";VM-.095*FX,.1*FY;"LB";TS(I);""+CHR$(3)
3080 NEXT I
3090 PRINT #1,"PUSH;PA0,OSP0;"
3100 CLOSE:END

```

4.2 ARRANAL.BAS Listing

```

10 'ARRANAL, PASP CONCENTRATOR ARRAY ACCEPTANCE TEST DATA ANALYSIS PROGRAM---
20 'CITP=MINICASSEGRAINIAN ISC TILT POSITIVE
30 'CITPC=MINICASSEGRAINIAN ISC TILT POSITIVE CORRECTED
40 'CVTP=MINICASSEGRAINIAN VOC TILT POSITIVE
50 'CVTPC=MINICASSEGRAINIAN VOC TILT POSITIVE CORRECTED
60 'CPTP=MINICASSEGRAINIAN PMX TILT POSITIVE
70 'CPTPC=MINICASSEGRAINIAN PMX TILT POSITIVE CORRECTED
80 'CRTP=MINICASSEGRAINIAN REF TILT POSITIVE
90 'CITN=MINICASSEGRAINIAN ISC TILT NEGATIVE
100 'CITNC=MINICASSEGRAINIAN ISC TILT NEGATIVE CORRECTED
110 'CVTN=MINICASSEGRAINIAN VOC TILT NEGATIVE
120 'CVTNC=MINICASSEGRAINIAN VOC TILT NEGATIVE CORRECTED
130 'CPTN=MINICASSEGRAINIAN PMX TILT NEGATIVE
140 'CPTNC=MINICASSEGRAINIAN PMX TILT NEGATIVE CORRECTED
150 'CRTN=MINICASSEGRAINIAN REF TILT NEGATIVE
160 'CIRP=MINICASSEGRAINIAN ISC ROTATE POSITIVE
170 'CIRPC=MINICASSEGRAINIAN ISC ROTATE POSITIVE CORRECTED
180 'CVRP=MINICASSEGRAINIAN VOC ROTATE POSITIVE
190 'CVRPC=MINICASSEGRAINIAN VOC ROTATE POSITIVE CORRECTED
200 'CPRP=MINICASSEGRAINIAN PMX ROTATE POSITIVE

```

```

210 'CPRPC=MINICASSEGRAINIAN PMX ROTATE POSITIVE CORRECTED
220 'CRRP=MINICASSEGRAINIAN REF ROTATE POSITIVE
230 'CIRN=MINICASSEGRAINIAN ISC ROTATE NEGATIVE
240 'CIRNC=MINICASSEGRAINIAN ISC ROTATE NEGATIVE CORRECTED
250 'CVRN=MINICASSEGRAINIAN VOC ROTATE NEGATIVE
260 'CVRNC=MINICASSEGRAINIAN VOC ROTATE NEGATIVE CORRECTED
270 'CPRN=MINICASSEGRAINIAN PMX ROTATE NEGATIVE
280 'CPRNC=MINICASSEGRAINIAN PMX ROTATE NEGATIVE CORRECTED
290 'CRRN=MINICASSEGRAINIAN REF ROTATE NEGATIVE
300 'CIF=MINICASSEGRAINIAN ISC FINISH
310 'CIFC=MINICASSEGRAINIAN ISC FINISH CORRECTED
320 'CVF=MINICASSEGRAINIAN VOC FINISH
330 'CVFC=MINICASSEGRAINIAN VOC FINISH CORRECTED
340 'CPF=MINICASSEGRAINIAN PMX FINISH
350 'CPFC=MINICASSEGRAINIAN PMX FINISH CORRECTED
360 'CRF=MINICASSEGRAINIAN REF FINISH
370 'SITP=SLATS ISC TILT POSITIVE
380 'SITPC=SLATS ISC TILT POSITIVE CORRECTED
390 'SVTP=SLATS VOC TILT POSITIVE
400 'SVTPC=SLATS VOC TILT POSITIVE CORRECTED
410 'SPTP=SLATS PMX TILT POSITIVE
420 'SPTPC=SLATS PMX TILT POSITIVE CORRECTED
430 'SRTP=SLATS REF TILT POSITIVE
440 'SITN=SLATS ISC TILT NEGATIVE
450 'SITNC=SLATS ISC TILT NEGATIVE CORRECTED
460 'SVTN=SLATS VOC TILT NEGATIVE
470 'SVTNC=SLATS VOC TILT NEGATIVE CORRECTED
480 'SPTN=SLATS PMX TILT NEGATIVE
490 'SPTNC=SLATS PMX TILT NEGATIVE CORRECTED
500 'SRTN=SLATS REF TILT NEGATIVE
510 'SIRP=SLATS ISC ROTATE POSITIVE
520 'SIRPC=SLATS ISC ROTATE POSITIVE CORRECTED
530 'SVRP=SLATS VOC ROTATE POSITIVE
540 'SVRPC=SLATS VOC ROTATE POSITIVE CORRECTED
550 'SPRP=SLATS PMX ROTATE POSITIVE
560 'SPRPC=SLATS PMX ROTATE POSITIVE CORRECTED
570 'SRRP=SLATS REF ROTATE POSITIVE
580 'SIRN=SLATS ISC ROTATE NEGATIVE
590 'SIRNC=SLATS ISC ROTATE NEGATIVE CORRECTED
600 'SVRN=SLATS VOC ROTATE NEGATIVE
610 'SVRNC=SLATS VOC ROTATE NEGATIVE CORRECTED
620 'SPRN=SLATS PMX ROTATE NEGATIVE
630 'SPRNC=SLATS PMX ROTATE NEGATIVE CORRECTED
640 'SRRN=SLATS REF ROTATE NEGATIVE
650 'SIF=SLATS ISC FINISH
660 'SIFC=SLATS ISC FINISH CORRECTED
670 'SVF=SLATS VOC FINISH
680 'SVFC=SLATS VOC FINISH CORRECTED
690 'SPF=SLATS PMX FINISH
700 'SPFC=SLATS PMX FINISH CORRECTED
710 'SRF=SLATS REF FINISH
720 DIM CITP(20),CITPC(20),CVTP(20),CVTPC(20),CPTP(20),CPTPC(20)
730 DIM CITN(20),CITNC(20),CVTN(20),CVTNC(20),CPTN(20),CPTNC(20)
740 DIM CIRP(20),CIRPC(20),CVRP(20),CVRPC(20),CPRP(20),CPRPC(20)
750 DIM CIRN(20),CIRNC(20),CVRN(20),CVRNC(20),CPRN(20),CPRNC(20)
760 DIM SITP(20),SITPC(20),SVTP(20),SVTPC(20),SPTP(20),SPTPC(20)
770 DIM SITN(20),SITNC(20),SVTN(20),SVTNC(20),SPTN(20),SPTNC(20)
780 DIM SIRP(20),SIRPC(20),SVRP(20),SVRPC(20),SPRP(20),SPRPC(20)
790 DIM SIRN(20),SIRNC(20),SVRN(20),SVRNC(20),SPRN(20),SPRNC(20)
800 DIM CRTP(20),CRTN(20),CRRP(20),CRRN(20)
810 DIM SRTP(20),SRTN(20),SRRP(20),SRRN(20)
820 A$="TILT POSITIVE":B$="TILT NEGATIVE":C$="ROTATION POSITIVE":D$="ROTATION NEGATIVE":E$="MINICAS":F$="SLATS":LPRINT
CHR$(27)"I"CHR$(5)
830 SCREEN 0:COLOR 7,1:KEY OFF:CLS:PRINT TAB(35)"** MENU **"
840 PRINT" A. ENTER MINICAS TILT POS DATA"
850 PRINT" B. ENTER MINICAS TILT NEG DATA"
860 PRINT" C. ENTER MINICAS ROTA POS DATA"
870 PRINT" D. ENTER MINICAS ROTA NEG DATA"
880 PRINT" E. ENTER SLATS TILT POS DATA"
890 PRINT" F. ENTER SLATS TILT NEG DATA"
900 PRINT" G. ENTER SLATS ROTA POS DATA"
910 PRINT" H. ENTER SLATS ROTA NEG DATA"
920 PRINT" I. DISPLAY MINICAS TILT POS DATA"

```

```

930 PRINT" J. DISPLAY MINICAS TILT NEG DATA"
940 PRINT" K. DISPLAY MINICAS ROTA POS DATA"
950 PRINT" L. DISPLAY MINICAS ROTA NEG DATA"
960 PRINT" M. DISPLAY SLATS TILT POS DATA"
970 PRINT" N. DISPLAY SLATS TILT NEG DATA"
980 PRINT" O. DISPLAY SLATS ROTA POS DATA"
990 PRINT" P. DISPLAY SLATS ROTA NEG DATA"
1000 PRINT" Q. PERFORM INTENSITY CORRECTIONS TO AMO"
1010 PRINT" R. READ DATA FROM DISK FILE"
1020 PRINT" S. SAVE DATA TO DISK FILE"
1030 PRINT" T. DISPLAY NORMAL DATA ANALYSIS"
1040 PRINT" U. GO TO PLOT MENU"
1050 PRINT" V. QUIT"
1060 PRINT TAB(29)"...SELECT A LETTER..."
1070 Z$=INPUT$(1)
1080 IF Z$="A" THEN 1310
1090 IF Z$="B" THEN 1420
1100 IF Z$="C" THEN 1530
1110 IF Z$="D" THEN 1640
1120 IF Z$="E" THEN 1780
1130 IF Z$="F" THEN 1860
1140 IF Z$="G" THEN 1940
1150 IF Z$="H" THEN 2050
1160 IF Z$="I" THEN 3010
1170 IF Z$="J" THEN 3010
1180 IF Z$="K" THEN 3010
1190 IF Z$="L" THEN 3010
1200 IF Z$="M" THEN 3090
1210 IF Z$="N" THEN 3090
1220 IF Z$="O" THEN 3130
1230 IF Z$="P" THEN 3130
1240 IF Z$="Q" THEN 2190
1250 IF Z$="R" THEN 2750
1260 IF Z$="S" THEN 2490
1270 IF Z$="T" THEN 3210
1280 IF Z$="U" THEN 3770
1290 IF Z$="V" THEN END
1300 '-----ENTER RAW DATA-----
1310 CLS:PRINT
1320 FOR I=0 TO 15
1330 IF I=0 THEN PRINT E$; "A$" ANGLE = 0":GOTO 1380
1340 IF I=13 THEN PRINT E$; "A$" ANGLE = 3.5":GOTO 1380
1350 IF I=14 THEN PRINT E$; "A$" ANGLE = 4.0":GOTO 1380
1360 IF I=15 THEN PRINT E$; "A$" ANGLE = 5.0":GOTO 1380
1370 PRINT E$; "A$" ANGLE =";1/4
1380 INPUT" ISC,VOC,IPM,VPM,REF ";A,B,C,D,E
1390 C1P(I)=A:CV1P(I)=B:C1PT(I)=C*D:CR1P(I)=E
1400 NEXT
1410 GOTO 830
1420 CLS
1430 FOR I=0 TO 15
1440 IF I=0 THEN PRINT E$; "B$" ANGLE = 0":GOTO 1490
1450 IF I=13 THEN PRINT E$; "B$" ANGLE = 3.5":GOTO 1490
1460 IF I=14 THEN PRINT E$; "B$" ANGLE = 4.0":GOTO 1490
1470 IF I=15 THEN PRINT E$; "B$" ANGLE = 5.0":GOTO 1490
1480 PRINT E$; "B$" ANGLE =";1/4
1490 INPUT" ISC,VOC,IPM,VPM,REF ";A,B,C,D,E
1500 C1TN(I)=A:CV1TN(I)=B:C1PTN(I)=C*D:CR1TN(I)=E
1510 NEXT
1520 GOTO 830
1530 CLS
1540 FOR I=0 TO 15
1550 IF I=0 THEN PRINT E$; "C$" ANGLE = 0":GOTO 1600
1560 IF I=13 THEN PRINT E$; "C$" ANGLE = 3.5":GOTO 1600
1570 IF I=14 THEN PRINT E$; "C$" ANGLE = 4.0":GOTO 1600
1580 IF I=15 THEN PRINT E$; "C$" ANGLE = 5.0":GOTO 1600
1590 PRINT E$; "C$" ANGLE =";1/4
1600 INPUT" ISC,VOC,IPM,VPM,REF ";A,B,C,D,E
1610 C1RP(I)=A:CV1RP(I)=B:C1PRP(I)=C*D:CR1RP(I)=E
1620 NEXT
1630 GOTO 830
1640 CLS
1650 FOR I=0 TO 15

```

```

1660 IF I=0 THEN PRINT E$"; "D$" ANGLE = 0":GOTO 1710
1670 IF I=13 THEN PRINT E$"; "D$" ANGLE = 3.5":GOTO 1710
1680 IF I=14 THEN PRINT E$"; "D$" ANGLE = 4.0":GOTO 1710
1690 IF I=15 THEN PRINT E$"; "D$" ANGLE = 5.0":GOTO 1710
1700 PRINT E$"; "D$" ANGLE ="I/4
1710 INPUT" ISC,VOC,IPM,VPM,REF ";A,B,C,D,E
1720 CIRN(I)=A:CVRN(I)=B:CPRN(I)=C*D:CRRN(I)=E
1730 NEXT
1740 PRINT E$"; FINISH ANGLE = 0"
1750 INPUT" ISC,VOC,IPM,VPM,REF ";A,B,C,D,E
1760 CIF=A:CVF=B:CPF=C*D:CRF=E
1770 GOTO 830
1780 CLS
1790 FOR I=0 TO 10
1800 IF I=0 THEN PRINT F$"; "A$" ANGLE = 0":GOTO 1820
1810 PRINT F$"; "A$" ANGLE ="I/4
1820 INPUT" ISC,VOC,IPM,VPM,REF ";A,B,C,D,E
1830 SITP(I)=A:SVTP(I)=B:SPTP(I)=C*D:SRTP(I)=E
1840 NEXT
1850 GOTO 830
1860 CLS
1870 FOR I=0 TO 10
1880 IF I=0 THEN PRINT F$"; "B$" ANGLE = 0":GOTO 1900
1890 PRINT F$"; "B$" ANGLE ="I/4
1900 INPUT" ISC,VOC,IPM,VPM,REF ";A,B,C,D,E
1910 SITN(I)=A:SVTN(I)=B:SPTN(I)=C*D:SRTN(I)=E
1920 NEXT
1930 GOTO 830
1940 CLS
1950 FOR I=0 TO 9
1960 IF I=0 THEN PRINT F$"; "C$" ANGLE = 0":GOTO 2010
1970 IF I=7 THEN PRINT F$"; "C$" ANGLE = 10":GOTO 2010
1980 IF I=8 THEN PRINT F$"; "C$" ANGLE = 20":GOTO 2010
1990 IF I=9 THEN PRINT F$"; "C$" ANGLE = 30":GOTO 2010
2000 PRINT F$"; "C$" ANGLE ="I
2010 INPUT" ISC,VOC,IPM,VPM,REF ";A,B,C,D,E
2020 SIRP(I)=A:SVRP(I)=B:SPRP(I)=C*D:SRRP(I)=E
2030 NEXT
2040 GOTO 830
2050 CLS
2060 FOR I=0 TO 9
2070 IF I=0 THEN PRINT F$"; "D$" ANGLE = 0":GOTO 2120
2080 IF I=7 THEN PRINT F$"; "D$" ANGLE = 10":GOTO 2120
2090 IF I=8 THEN PRINT F$"; "D$" ANGLE = 20":GOTO 2120
2100 IF I=9 THEN PRINT F$"; "D$" ANGLE = 30":GOTO 2120
2110 PRINT F$"; "D$" ANGLE ="I
2120 INPUT" ISC,VOC,IPM,VPM,REF ";A,B,C,D,E
2130 SIRN(I)=A:SVRN(I)=B:SPRN(I)=C*D:SRRN(I)=E
2140 NEXT
2150 PRINT F$"; FINISH ANGLE = 0"
2160 INPUT" ISC,VOC,IPM,VPM,REF ";A,B,C,D,E
2170 SIF=A:SVF=B:SPF=C*D:SRF=E
2180 GOTO 830
2190 '-----CORRECT DATA TO AMO INTENSITY-----
2200 CLS:PRINT:PRINT TAB(24)"...COMPUTING DATA CORRECTIONS..."
2210 A=59.9:B=1.067
2220 FOR I=0 TO 15
2230 CITPC(I)=CITP(I)*A/C RTP(I):CVTPC(I)=B*CVTP(I):CPTPC(I)=CPTP(I)*A*B/C RTP(I)
2240 CITNC(I)=CITN(I)*A/C RTN(I):CVTNC(I)=B*CVTN(I):CPTNC(I)=CPTN(I)*A*B/C RTN(I)
2250 CIRPC(I)=CIRP(I)*A/CRRP(I):CVRPC(I)=B*CVRP(I):CPRPC(I)=CPRP(I)*A*B/CRRP(I)
2260 CIRNC(I)=CIRN(I)*A/CRRN(I):CVRNC(I)=B*CVRN(I):CPRNC(I)=CPRN(I)*A*B/CRRN(I)
2270 NEXT
2280 CIFC=CIF*A/CRF:CVFC=B*CVF:CPFC=CPF*A*B/CRF
2290 FOR I=0 TO 10
2300 SITPC(I)=SITP(I)*A/S RTP(I):SVTPC(I)=B*SVTP(I):SPTPC(I)=SPTP(I)*A*B/S RTP(I)
2310 SITNC(I)=SITN(I)*A/S RTN(I):SVTNC(I)=B*SVTN(I):SPTNC(I)=SPTN(I)*A*B/S RTN(I)
2320 NEXT
2330 FOR I=0 TO 9
2340 RR=SRRP(I)
2350 IF I=7 THEN RR=SRRP(I)/COS(10*3.14159/180)
2360 IF I=8 THEN RR=SRRP(I)/COS(20*3.14159/180)
2370 IF I=9 THEN RR=SRRP(I)/COS(30*3.14159/180)
2380 SIRPC(I)=SIRP(I)*A/RR:SVRPC(I)=B*SVRP(I):SPRPC(I)=SPRP(I)*A*B/RR

```



```

2390 NEXT
2400 FOR I=0 TO 9
2410 RR=SRRN(I)
2420 IF I=7 THEN RR=SRRN(I)/COS(10*3.14159/180)
2430 IF I=8 THEN RR=SRRN(I)/COS(20*3.14159/180)
2440 IF I=9 THEN RR=SRRN(I)/COS(30*3.14159/180)
2450 SIRNC(I)=SIRN(I)*A/RR:SVRNC(I)=B*SVRN(I):SPRNC(I)=SPRN(I)*A*B/RR
2460 NEXT
2470 SIFC=SIF*A/SRF:SVFC=B*SVF:SPFC=SPF*A*B/SRF
2480 GOTO 830
2490 '-----SAVE DATA TO DISK-----
2500 PRINT:PRINT TAB(20)"...SAVING DATA FILE 'ARRDATA' TO DISK..."
2510 OPEN "ARRDATA" FOR OUTPUT AS #1:GOSUB 2570
2520 PRINT:INPUT"DO YOU WANT TO SAVE A BACKUP DATA FILE (Y/N) ";Z$
2530 IF Z$<>"Y" AND Z$<>"N" THEN 2520
2540 IF Z$="N" THEN 2740
2550 PRINT:PRINT TAB(20)"...SAVING DATA FILE 'ARRDATAB' TO DISK..."
2560 OPEN "ARRDATAB" FOR OUTPUT AS #1:GOSUB 2570:GOTO 2740
2570 FOR I=0 TO 15
2580 PRINT #1,CITP(I);CVTP(I);CPTP(I);CRTP(I);CITN(I);CVTN(I);CPTN(I);CRTN(I);
2590 PRINT #1,CIRP(I);CVRP(I);CPRP(I);CRRP(I);CIRN(I);CVRN(I);CPRN(I);CRRN(I);
2600 PRINT #1,CITPC(I);CVTPC(I);CPTPC(I);CRTPC(I);CITNC(I);CVTNC(I);CPTNC(I);
2610 PRINT #1,CIRPC(I);CVRPC(I);CPRPC(I);CIRNC(I);CVRNC(I);CPRNC(I);
2620 NEXT
2630 PRINT #1,CIF;CIFC;CVF;CVFC;CPF;CPFC;CRF;
2640 FOR I=C TO 10
2650 PRINT #1,SITP(I);SVTP(I);SPTP(I);SRTP(I);SITN(I);SVTN(I);SPTN(I);SRTN(I);
2660 PRINT #1,SITPC(I);SVTPC(I);SPTPC(I);SITNC(I);SVTNC(I);SPTNC(I);
2670 NEXT
2680 FOR I=0 TO 9
2690 PRINT #1,SIRP(I);SVRP(I);SPRP(I);SRRP(I);SIRN(I);SVRN(I);SPRN(I);SRRN(I);
2700 PRINT #1,SIRPC(I);SVRPC(I);SPRPC(I);SIRNC(I);SVRNC(I);SPRNC(I);
2710 NEXT
2720 PRINT #1,SIF;SIFC;SVF;SVFC;SPF;SPFC;SRF;
2730 CLOSE:RETURN
2740 GOTO 830
2750 '-----READ DATA FILE FROM DISK-----
2760 CLS:PRINT
2770 INPUT"READ 'ARRDATA' (1) OR 'ARRDATAB' (2) DATA FILE ";Z
2780 IF Z<1 AND Z>2 THEN 2770
2790 IF Z=1 THEN N$="ARRDATA" ELSE N$="ARRDATAB"
2800 PRINT:PRINT TAB(21)"...READING DISK DATA FILE '"N$"'..."
2810 OPEN N$ FOR INPUT AS #1
2820 FOR I=0 TO 15
2830 INPUT #1,CITP(I),CVTP(I),CPTP(I),CRTP(I),CITN(I),CVTN(I),CPTN(I),CRTN(I)
2840 INPUT #1,CIRP(I),CVRP(I),CPRP(I),CRRP(I),CIRN(I),CVRN(I),CPRN(I),CRRN(I)
2850 INPUT #1,CITPC(I),CVTPC(I),CPTPC(I),CRTPC(I),CITNC(I),CVTNC(I),CPTNC(I)
2860 INPUT #1,CIRPC(I),CVRPC(I),CPRPC(I),CIRNC(I),CVRNC(I),CPRNC(I)
2870 NEXT
2880 INPUT #1,CIF,CIFC,CVF,CVFC,CPF,CPFC,CRF
2890 FOR I=0 TO 10
2900 INPUT #1,SITP(I),SVTP(I),SPTP(I),SRTP(I),SITN(I),SVTN(I),SPTN(I),SRTN(I)
2910 INPUT #1,SITPC(I),SVTPC(I),SPTPC(I),SITNC(I),SVTNC(I),SPTNC(I)
2920 NEXT
2930 FOR I=0 TO 9
2940 INPUT #1,SIRP(I),SVRP(I),SPRP(I),SRRP(I),SIRN(I),SVRN(I),SPRN(I),SRRN(I)
2950 INPUT #1,SIRPC(I),SVRPC(I),SPRPC(I),SIRNC(I),SVRNC(I),SPRNC(I)
2960 NEXT
2970 INPUT #1,SIF,SIFC,SVF,SVFC,SPF,SPFC,SRF
2980 CLOSE
2990 GOTO 830
3000 '-----DISPLAY DATA-----
3010 FOR I=0 TO 15
3020 IF Z$="I" THEN PRINT CITP(I)TAB(10)CVTP(I)TAB(20)CPTP(I)TAB(30)CRTP(I)TAB(40)CITPC(I)TAB(50)CVTPC(I)TAB(60)CPTPC(I)
3030 IF Z$="J" THEN PRINT CITN(I)TAB(10)CVTN(I)TAB(20)CPTN(I)TAB(30)CRTN(I)TAB(40)CITNC(I)TAB(50)CVTNC(I)TAB(60)CPTNC(I)
3040 IF Z$="K" THEN PRINT CIRP(I)TAB(10)CVRP(I)TAB(20)CPRP(I)TAB(30)CRRP(I)TAB(40)CIRPC(I)TAB(50)CVRPC(I)TAB(60)CPRPC(I)
3050 IF Z$="L" THEN PRINT CIRN(I)TAB(10)CVRN(I)TAB(20)CPRN(I)TAB(30)CRRN(I)TAB(40)CIRNC(I)TAB(50)CVRNC(I)TAB(60)CPRNC(I)
3060 NEXT
3070 IF Z$="M" THEN PRINT:PRINT CIF;TAB(10)CVF;TAB(20)CPF;TAB(30)CRF;TAB(40)CIFC;TAB(50)CVFC;TAB(60)CPFC
3080 GOTO 3180
3090 FOR I=0 TO 10
3100 IF Z$="N" THEN PRINT SITP(I)TAB(10)SVTP(I)TAB(20)SPTP(I)TAB(30)SRTP(I)TAB(40)SITPC(I)TAB(50)SVTPC(I)TAB(60)SPTPC(I)
3110 IF Z$="O" THEN PRINT SITN(I)TAB(10)SVTN(I)TAB(20)SPTN(I)TAB(30)SRTN(I)TAB(40)SITNC(I)TAB(50)SVTNC(I)TAB(60)SPTNC(I)

```

```

3120 NEXT
3130 FOR I=0 TO 9
3140 IF Z$="O" THEN PRINT SIRP(I)TAB(10)SVRP(I)TAB(20)SPRP(I)TAB(30)SRRP(I)TAB(40)SIRPC(I)TAB(50)SVRPC(I)TAB(60)SPRPC(I)
3150 IF Z$="P" THEN PRINT SIRN(I)TAB(10)SVRN(I)TAB(20)SPRN(I)TAB(30)SRRN(I)TAB(40)SIRNC(I)TAB(50)SVRNC(I)TAB(60)SPRNC(I)
3160 NEXT
3170 IF Z$="P" THEN PRINT:PRINT SIF;TAB(10)SVF;TAB(20)SPF;TAB(30)SRF;TAB(40)SIFC;TAB(50)SVFC;TAB(60)SPFC
3180 PRINT:PRINT"PRESS ANY KEY TO CONTINUE"
3190 Z$=INPUT$(1):IF Z$="" THEN 3190
3200 GOTO 830
3210 '-----DISPLAY NORMAL ANALYSIS-----
3220 CINA=(CITPC(0)+CITNC(0)+CIRPC(0)+CIRNC(0)+CIFC)/5:CINH=0
3230 IF CITPC(0)>CINH THEN CINH=CITPC(0)
3240 IF CITNC(0)>CINH THEN CINH=CITNC(0)
3250 IF CIRPC(0)>CINH THEN CINH=CIRPC(0)
3260 IF CIRNC(0)>CINH THEN CINH=CIRNC(0)
3270 IF CIFC>CINH THEN CINH=CIFC
3280 CINL=CINH
3290 IF CITPC(0)<CINL THEN CINL=CITPC(0)
3300 IF CITNC(0)<CINL THEN CINL=CITNC(0)
3310 IF CIRPC(0)<CINL THEN CINL=CIRPC(0)
3320 IF CIRNC(0)<CINL THEN CINL=CIRNC(0)
3330 IF CIFC<CINL THEN CINL=CIFC
3340 CPNA=(CPTPC(0)+CPTNC(0)+CPRPC(0)+CPRNC(0)+CPFC)/5:CPNH=0
3350 IF CPTPC(0)>CPNH THEN CPNH=CPTPC(0)
3360 IF CPTNC(0)>CPNH THEN CPNH=CPTNC(0)
3370 IF CPRPC(0)>CPNH THEN CPNH=CPRPC(0)
3380 IF CPRNC(0)>CPNH THEN CPNH=CPRNC(0)
3390 IF CPFC>CPNH THEN CPNH=CPFC
3400 CPNL=CPNH
3410 IF CPTPC(0)<CPNL THEN CPNL=CPTPC(0)
3420 IF CPTNC(0)<CPNL THEN CPNL=CPTNC(0)
3430 IF CPRPC(0)<CPNL THEN CPNL=CPRPC(0)
3440 IF CPRNC(0)<CPNL THEN CPNL=CPRNC(0)
3450 IF CPFC<CPNL THEN CPNL=CPFC
3460 PRINT CINA,CINH,CINL,(CINH-CINA)*100/CINA,(CINL-CINA)*100/CINA
3470 PRINT CPNA,CPNH,CPNL,(CPNH-CPNA)*100/CPNA,(CPNL-CPNA)*100/CPNA
3480 SINA=(SITPC(0)+SITNC(0)+SIRPC(0)+SIRNC(0)+SIFC)/5:SINH=0
3490 IF SITPC(0)>SINH THEN SINH=SITPC(0)
3500 IF SITNC(0)>SINH THEN SINH=SITNC(0)
3510 IF SIRPC(0)>SINH THEN SINH=SIRPC(0)
3520 IF SIRNC(0)>SINH THEN SINH=SIRNC(0)
3530 IF SIFC>SINH THEN SINH=SIFC
3540 SINL=SINH
3550 IF SITPC(0)<SINL THEN SINL=SITPC(0)
3560 IF SITNC(0)<SINL THEN SINL=SITNC(0)
3570 IF SIRPC(0)<SINL THEN SINL=SIRPC(0)
3580 IF SIRNC(0)<SINL THEN SINL=SIRNC(0)
3590 IF SIFC<SINL THEN SINL=SIFC
3600 SPNA=(SPTPC(0)+SPTNC(0)+SPRPC(0)+SPRNC(0)+SPFC)/5:SPNH=0
3610 IF SPTPC(0)>SPNH THEN SPNH=SPTPC(0)
3620 IF SPTNC(0)>SPNH THEN SPNH=SPTNC(0)
3630 IF SPRPC(0)>SPNH THEN SPNH=SPRPC(0)
3640 IF SPRNC(0)>SPNH THEN SPNH=SPRNC(0)
3650 IF SPFC>SPNH THEN SPNH=SPFC
3660 SPNL=SPNH
3670 IF SPTPC(0)<SPNL THEN SPNL=SPTPC(0)
3680 IF SPTNC(0)<SPNL THEN SPNL=SPTNC(0)
3690 IF SPRPC(0)<SPNL THEN SPNL=SPRPC(0)
3700 IF SPRNC(0)<SPNL THEN SPNL=SPRNC(0)
3710 IF SPFC<SPNL THEN SPNL=SPFC
3720 PRINT SINA,SINH,SINL,(SINH-SINA)*100/SINA,(SINL-SINA)*100/SINA
3730 PRINT SPNA,SPNH,SPNL,(SPNH-SPNA)*100/SPNA,(SPNL-SPNA)*100/SPNA
3740 PRINT:PRINT"PRESS ANY KEY TO CONTINUE"
3750 Z$=INPUT$(1):IF Z$="" THEN 3750
3760 GOTO 830
3770 '-----DISPLAY DATA PLOTS-----
3780 SCREEN 2:CLS
3790 LOS="NORMALIZED OUTPUT":LAS="DEGREES OFF NORMAL"
3800 PRINT:PRINT TAB(35)"** MENU **":PRINT
3810 PRINT" A. PLOT MINICAS TILT AXIS"
3820 PRINT" B. PLOT MINICAS ROTATION AXIS"
3830 PRINT" C. PLOT SLATS TILT AXIS"
3840 PRINT" D. PLOT SLATS ROTATION AXIS"

```

```

3850 PRINT" E. PRINT OUTPUT DATA"
3860 PRINT" F. RETURN TO MAIN MENU"
3870 PRINT:PRINT TAB(29)"...SELECT A LETTER..."
3880 Z$=INPUT$(1)
3890 IF Z$="A" THEN X=6:GOSUB 3950:GOTO 4070
3900 IF Z$="R" THEN X=6:GOSUB 3950:GOTO 4340
3910 IF Z$="C" THEN X=3:GOSUB 3950:GOTO 4610
3920 IF Z$="D" THEN X=30:GOSUB 3950:GOTO 4760
3930 IF Z$="E" THEN SCREEN 0:COLOR 7,1:CLS:GOTO 5030
3940 IF Z$="F" THEN SCREEN 0:GOTO 830
3950 VIEW:CLS
3960 VIEW (1,1)-(638,198),,1
3970 WINDOW (-X-X/6,-.25)-(X+X/6,1.1)
3980 LINE (-X,0)-(X,0):LINE (0,0)-(0,1.1)
3990 FOR I=.1 TO 1.1 STEP .1:LINE (-X/60,I)-(X/60,I):NEXT
4000 FOR I=-X TO X STEP X/6:LINE (I,.02)-(I,-.02):NEXT
4010 FOR I=-X TO X STEP X/6:LOCATE 22,40+80*I*6/(X*14):PRINT I:NEXT
4020 LOCATE 23,33:PRINT LAS
4030 FOR I=3 TO 19:LOCATE I,3:PRINT MID$(LO$,I-2,1):NEXT
4040 LINE (-X,1)-(X,1),,,&HCCCC:LOCATE 2,7:PRINT"1.0"
4050 LINE (-X,.5)-(X,.5),,,&HCCCC:LOCATE 11,7:PRINT"0.5"
4060 RETURN
4070 PN=CPTPC(0):PSET (0,1)
4080 FOR I=1 TO 12:LINE -(I*.25,CPTPC(I)/PN):NEXT
4090 LINE -(3.5,CPTPC(13)/PN)
4100 LINE -(4,CPTPC(14)/PN)
4110 LINE -(5,CPTPC(15)/PN)
4120 PN=CPTNC(0):PSET (0,1)
4130 FOR I=1 TO 12:LINE -(I*.25,CPTNC(I)/PN):NEXT
4140 LINE -(-3.5,CPTNC(13)/PN)
4150 LINE -(-4,CPTNC(14)/PN)
4160 LINE -(-5,CPTNC(15)/PN)
4170 PN=CITPC(0):PSET (0,1)
4180 FOR I=1 TO 12:LINE -(I*.25,CITPC(I)/PN):NEXT
4190 LINE -(3.5,CITPC(13)/PN)
4200 LINE -(4,CITPC(14)/PN)
4210 LINE -(5,CITPC(15)/PN)
4220 PN=CITNC(0):PSET (0,1)
4230 FOR I=1 TO 12:LINE -(I*.25,CITNC(I)/PN):NEXT
4240 LINE -(-3.5,CITNC(13)/PN)
4250 LINE -(-4,CITNC(14)/PN)
4260 LINE -(-5,CITNC(15)/PN)
4270 LOCATE 4,58:PRINT"UPPER CURVE = ISC":LOCATE 5,58:PRINT"LOWER CURVE = PMX"
4280 LOCATE 24,16:PRINT "FIGURE 8, MINI-CASSEGRAINIAN CONCENTRATOR TILT AXIS";
4290 GOSUB 6230
4300 LOCATE 1,1
4310 PRINT"PRESS ANY KEY TO CONTINUE"
4320 Z$=INPUT$(1):IF Z$="" THEN 4320
4330 VIEW:CLS:GOTO 3800
4340 PN=CPRPC(0):PSET (0,1)
4350 FOR I=1 TO 12:LINE -(I*.25,CPRPC(I)/PN):NEXT
4360 LINE -(3.5,CPRPC(13)/PN)
4370 LINE -(4,CPRPC(14)/PN)
4380 LINE -(5,CPRPC(15)/PN)
4390 PN=CPRNC(0):PSET (0,1)
4400 FOR I=1 TO 12:LINE -(I*.25,CPRNC(I)/PN):NEXT
4410 LINE -(-3.5,CPRNC(13)/PN)
4420 LINE -(-4,CPRNC(14)/PN)
4430 LINE -(-5,CPRNC(15)/PN)
4440 PN=CIRPC(0):PSET (0,1)
4450 FOR I=1 TO 12:LINE -(I*.25,CIRPC(I)/PN):NEXT
4460 LINE -(3.5,CIRPC(13)/PN)
4470 LINE -(4,CIRPC(14)/PN)
4480 LINE -(5,CIRPC(15)/PN)
4490 PN=CIRNC(0):PSET (0,1)
4500 FOR I=1 TO 12:LINE -(I*.25,CIRNC(I)/PN):NEXT
4510 LINE -(-3.5,CIRNC(13)/PN)
4520 LINE -(-4,CIRNC(14)/PN)
4530 LINE -(-5,CIRNC(15)/PN)
4540 LOCATE 4,58:PRINT"UPPER CURVE = ISC":LOCATE 5,58:PRINT"LOWER CURVE = PMX"
4550 LOCATE 24,14:PRINT "FIGURE 9, MINI-CASSEGRAINIAN CONCENTRATOR ROTATION AXIS";
4560 GOSUB 6230
4570 LOCATE 1,1

```

```

4580 PRINT"PRESS ANY KEY TO CONTINUE"
4590 Z$=INPUT$(1):IF Z$="" THEN 4590
4600 VIEW:CLS:GOTO 3800
4610 PN=SPTPC(0):PSET (0,1)
4620 FOR I=1 TO 10:LINE -(I*.25,SPTPC(I)/PN):NEXT
4630 PN=SPTNC(0):PSET (0,1)
4640 FOR I=1 TO 10:LINE -(I*.25,SPTNC(I)/PN):NEXT
4650 PN=SITPC(0):PSET (0,1)
4660 FOR I=1 TO 10:LINE -(I*.25,SITPC(I)/PN):NEXT
4670 PN=SITNC(0):PSET (0,1)
4680 FOR I=1 TO 10:LINE -(I*.25,SITNC(I)/PN):NEXT
4690 LOCATE 4,7:PRINT"UPPER CURVE = ISC":LOCATE 5,7:PRINT"LOWER CURVE = PMX"
4700 LOCATE 24,21:PRINT " FIGURE 11, SLATS CONCENTRATOR TILT AXIS";
4710 GOSUB 6230
4720 LOCATE 1,1
4730 PRINT"PRESS ANY KEY TO CONTINUE"
4740 Z$=INPUT$(1):IF Z$="" THEN 4740
4750 VIEW:CLS:GOTO 3800
4760 PN=SPRPC(0):PSET (0,1)
4770 FOR I=1 TO 6:LINE -(I,SPRPC(I)/PN):NEXT
4780 LINE -(10,SPRPC(7)/PN)
4790 LINE -(20,SPRPC(8)/PN)
4800 LINE -(30,SPRPC(9)/PN)
4810 PN=SPRNC(0):PSET (0,1)
4820 FOR I=1 TO 6:LINE -(I,SPRNC(I)/PN):NEXT
4830 LINE -(10,SPRNC(7)/PN)
4840 LINE -(20,SPRNC(8)/PN)
4850 LINE -(30,SPRNC(9)/PN)
4860 PN=SIRPC(0):PSET (0,1)
4870 FOR I=1 TO 6:LINE -(I,SIRPC(I)/PN):NEXT
4880 LINE -(10,SIRPC(7)/PN)
4890 LINE -(20,SIRPC(8)/PN)
4900 LINE -(30,SIRPC(9)/PN)
4910 PN=SIRNC(0):PSET (0,1)
4920 FOR I=1 TO 6:LINE -(I,SIRNC(I)/PN):NEXT
4930 LINE -(10,SIRNC(7)/PN)
4940 LINE -(20,SIRNC(8)/PN)
4950 LINE -(30,SIRNC(9)/PN)
4960 LOCATE 14,58:PRINT"UPPER CURVE = ISC":LOCATE 15,58:PRINT"LOWER CURVE = PMX"
4970 LOCATE 24,19:PRINT " FIGURE 12, SLATS CONCENTRATOR ROTATION AXIS";
4980 GOSUB 6230
4990 LOCATE 1,1
5000 PRINT"PRESS ANY KEY TO CONTINUE"
5010 Z$=INPUT$(1):IF Z$="" THEN 5010
5020 VIEW:CLS:GOTO 3800
5030 !-----PRINT ROUTINES-----
5040 PRINT:PRINT"WHEN SCREEN CLEARS, ACTIVATE 'Ctrl-Print Screen' WITHIN 4 SECONDS"
5050 PRINT:PRINT"PRESS ANY KEY WHEN READY TO CLEAR SCREEN"
5060 Z$=INPUT$(1):IF Z$="" THEN 5060
5070 CLS
5080 FOR I=1 TO 4000:NEXT
5090 PAS="MINI-CASSEGRAINIAN CONCENTRATOR ARRAY":PB$="SLATS CONCENTRATOR ARRAY"
5100 GOTO 5200
5110 FOR I=1 TO 6:PRINT:NEXT:CC=6
5120 CT=CT+1:PRINT TAB(23)"TABLE";CT;"- CONCENTRATOR ARRAY DATA":PRINT:CC=CC+2
5130 PRINT TAB(TNN)NN$:PRINT:CC=CC+2
5140 PRINT TAB(TN)N$:PRINT:CC=CC+2
5150 PRINT TAB(8)"ANGLE"TAB(18)"ISC"TAB(28)"VOC"TAB(38)"PMX"TAB(48)"ISC"TAB(58)"VOC"TAB(68)"PMX"
5160 PRINT TAB(9)"OFF"TAB(18)"RAW"TAB(28)"RAW"TAB(38)"RAW"TAB(48)"CORR"TAB(58)"CORR"TAB(68)"CORR"
5170 PRINT TAB(8)"NORM"TAB(18)"DATA"TAB(28)"DATA"TAB(38)"DATA"TAB(48)"AM0"TAB(58)"AM0"TAB(68)"AM0"
5180 PRINT TAB(8)"(DEG)"TAB(18)"(MA)"TAB(28)"(V)"TAB(38)"(MW)"TAB(48)"(MA)"TAB(58)"(V)"TAB(68)"(MW)":PRINT:CC=CC+5
5190 RETURN
5200 NN$=PAS:N$=A$:TNN=21:TN=33:GOSUB 5110
5210 FOR I=0 TO 15
5220 A=I:IF I=13 THEN A=14
5230 IF I=14 THEN A=16
5240 IF I=15 THEN A=20
5250 PRINT TAB(8)USING"###.###";A*.25;
5260 PRINT TAB(18)USING"###";CITP(I);
5270 PRINT TAB(27)USING"###.###";CVTP(I);
5280 PRINT TAB(37)USING"###";CPTP(I);
5290 PRINT TAB(48)USING"###";CITPC(I);
5300 PRINT TAB(57)USING"###.###";CVTPC(I);

```

```

5310 PRINT TAB(68)USING"####":CPTPC(I)
5320 NEXT:PRINT:PRINT:CC=CC+18
5330 PRINT TAB(TN)B$:PRINT:CC=CC+2
5340 FOR I=0 TO 15
5350 A=I:IF I=13 THEN A=14
5360 IF I=14 THEN A=16
5370 IF I=15 THEN A=20
5380 PRINT TAB(8)USING"###.###":A*-.25;
5390 PRINT TAB(18)USING"####":CITN(I);
5400 PRINT TAB(27)USING"###.###":CVTN(I);
5410 PRINT TAB(37)USING"####":CPTN(I);
5420 PRINT TAB(48)USING"####":CITNC(I);
5430 PRINT TAB(57)USING"###.###":CVTNC(I);
5440 PRINT TAB(68)USING"####":CPTNC(I)
5450 NEXT:PRINT:PRINT:CC=CC+18
5460 FOR I=1 TO 66-CC:PRINT:NEXT:CC=0
5470 NNS=PA$:N$=C$:TNN=21:TN=31:GOSUB 5110
5480 FOR I=0 TO 15
5490 A=I:IF I=13 THEN A=14
5500 IF I=14 THEN A=16
5510 IF I=15 THEN A=20
5520 PRINT TAB(8)USING"###.###":A*-.25;
5530 PRINT TAB(18)USING"####":CIRP(I);
5540 PRINT TAB(27)USING"###.###":CVRP(I);
5550 PRINT TAB(37)USING"####":CPRP(I);
5560 PRINT TAB(48)USING"####":CIRPC(I);
5570 PRINT TAB(57)USING"###.###":CVRPC(I);
5580 PRINT TAB(68)USING"####":CPRPC(I)
5590 NEXT:PRINT:PRINT:CC=CC+18
5600 PRINT TAB(31)D$:PRINT:CC=CC+2
5610 FOR I=0 TO 15
5620 A=I:IF I=13 THEN A=14
5630 IF I=14 THEN A=16
5640 IF I=15 THEN A=20
5650 PRINT TAB(8)USING"###.###":A*-.25;
5660 PRINT TAB(18)USING"####":CIRN(I);
5670 PRINT TAB(27)USING"###.###":CVRN(I);
5680 PRINT TAB(37)USING"####":CPRN(I);
5690 PRINT TAB(48)USING"####":CIRNC(I);
5700 PRINT TAB(57)USING"###.###":CVRNC(I);
5710 PRINT TAB(68)USING"####":CPRNC(I)
5720 NEXT:PRINT:PRINT:CC=CC+18
5730 FOR I=1 TO 66-CC:PRINT:NEXT:CC=0
5740 NNS=PB$:N$=A$:TNN=28:TN=33:GOSUB 5110
5750 FOR I=0 TO 10
5760 PRINT TAB(8)USING"###.###":I*-.25;
5770 PRINT TAB(18)USING"####":SITP(I);
5780 PRINT TAB(28)USING"###.###":SVTP(I);
5790 PRINT TAB(38)USING"####":SPTP(I);
5800 PRINT TAB(48)USING"####":SITPC(I);
5810 PRINT TAB(58)USING"###.###":SVTPC(I);
5820 PRINT TAB(68)USING"####":SPTPC(I)
5830 NEXT:PRINT:PRINT:CC=CC+13
5840 PRINT TAB(TN)B$:PRINT:CC=CC+2
5850 FOR I=0 TO 10
5860 PRINT TAB(8)USING"###.###":I*-.25;
5870 PRINT TAB(18)USING"####":SITN(I);
5880 PRINT TAB(28)USING"###.###":SVTN(I);
5890 PRINT TAB(38)USING"####":SPTN(I);
5900 PRINT TAB(48)USING"####":SITNC(I);
5910 PRINT TAB(58)USING"###.###":SVTNC(I);
5920 PRINT TAB(68)USING"####":SPTNC(I)
5930 NEXT:PRINT:PRINT:CC=CC+13
5940 FOR I=1 TO 66-CC:PRINT:NEXT:CC=0
5950 NNS=PB$:N$=C$:TNN=28:TN=31:GOSUB 5110
5960 FOR I=0 TO 9
5970 A=I:IF I=7 THEN A=10
5980 IF I=8 THEN A=20
5990 IF I=9 THEN A=30
6000 PRINT TAB(8)USING"###.###":A;
6010 PRINT TAB(18)USING"####":SIRP(I);
6020 PRINT TAB(28)USING"###.###":SVRP(I);
6030 PRINT TAB(38)USING"####":SPRP(I);

```

```

6040 PRINT TAB(48)USING"###";SIRPC(I);
6050 PRINT TAB(58)USING"###.#";SVRPC(I);
6060 PRINT TAB(68)USING"#####";SPRPC(I)
6070 NEXT:PRINT:PRINT:CC=CC+12
6080 PRINT TAB(TN)D$:PRINT:CC=CC+2
6090 FOR I=0 TO 9
6100 A=I:IF I=7 THEN A=10
6110 IF I=8 THEN A=20
6120 IF I=9 THEN A=30
6130 PRINT TAB(8)USING"###.#";-A;
6140 PRINT TAB(18)USING"#####";SIRN(I);
6150 PRINT TAB(28)USING"###.#";SVRN(I);
6160 PRINT TAB(38)USING"#####";SPRN(I);
6170 PRINT TAB(48)USING"#####";SIRNC(I);
6180 PRINT TAB(58)USING"###.#";SVRNC(I);
6190 PRINT TAB(68)USING"#####";SPRNC(I)
6200 NEXT:PRINT:PRINT:CC=CC+12
6210 FOR I=1 TO 66-CC:PRINT:NEXT:CC=0
6220 GOTO 830
6230 FOR I=1 TO 5:LPRINT:NEXT:LPRINT TAB(5)
6240 FOR I=1 TO 10000:NEXT
6250 RETURN

```

4.3 ORBITPP.BAS Listing

```

10 REM-----ORBITPP-----
20 DIM S(2901):P=3.141593/180:A$="SUN":B$="DARK":LPRINT CHR$(27)"I"CHR$(5)
30 SCREEN 1:COLOR 9,1:KEY OFF
40 GOSUB 260:GOSUB 150
50 AX=AA+AC*SA
60 ET=LI*(NM+1)+VP*DP*NM+VN*DN*NM
70 AT=AC*ET
80 AZ=AT+AX:AY=AX:GOSUB 500:GOSUB 710
90 AZ=AT+AD:AY=AD:GOSUB 500:GOSUB 710
100 AZ=AT+AE:AY=AE:GOSUB 500:GOSUB 710
110 PRINT#1,"VSPA0,OSPO,";CLOSE
120 END
130 PUT (-99,-99),S,PSET
140 END
150 VIEW:CLS
160 VIEW (46,2)-(275,193)
170 WINDOW (-99,-99)-(100,100)
180 CIRCLE (0,0),80
190 DRAW "TA=AB;NR92":DRAW "TA=AA;NR92"
200 LOCATE 4,21:PRINT"N"
210 LOCATE 22,21:PRINT"S"
220 LOCATE 15,16:PRINT A$
230 LOCATE 12,24:PRINT B$
240 GET (-99,-99)-(100,99),S
250 RETURN
260 REM-----MENU-----
270 CLS:PRINT"          MENU"
280 INPUT"NUM OF POS VOLTAGE STEPS ";VP
290 INPUT"POS DWELL (SECS) ";DP
300 INPUT"NUM OF NEG VOLTAGE STEPS ";VN
310 INPUT"NEG DWELL (SECS) ";DN
320 INPUT"NUM OF SCANS/ORBIT ";SC
330 INPUT"NUM OF MODULES TESTED/SCAN ";NM
340 INPUT"SUN ACQUISITION TIME (SECS) ";SA
350 INPUT"LP AND IV SCAN TIME (SECS) ";LI
360 AA=97.5:AB=309.5:AC=1/15:AD=168.17:AE=238.84
370 RETURN
380 REM-----PEN PLOT-----
390 R=30:CX=50:CY=37.5
400 OPEN "COM1:2400,S,7,1,RS,CS65535,DS,CD" AS #1
410 PRINT#1,"INIP1120,1155,9290,7280SCO,100,0,75;"
420 PRINT#1,"SP1PA";CX,CY;"CI";R;"1;"
430 PRINT#1,"PA";CX,CY;"VS5EW";R,AA,AB-AA-360;"
440 PRINT#1,"SR2,4PA49,64LBN"+CHR$(3)
450 PRINT#1,"SR2,4PA49,8LBS"+CHR$(3)
460 PRINT#1,"SR2,4PA35,31LBSUN"+CHR$(3)
470 PRINT#1,"SR2,4PA60,40LBDARK"+CHR$(3)

```

```

480 'PRINT#1,"VSPA0,OSP0;":CLOSE
490 RETURN
500 REM-----DRAW SEQUENCE-----
510 'LPRINT AY,AZ
520 X=88*COS(AY*P):Y=88*SIN(AY*P):PSET (X,Y)
530 FOR I=AY TO AZ STEP 2
540 X=88*COS(I*P):Y=88*SIN(I*P)
550 LINE -(X,Y)
560 NEXT
570 X=96*COS(AY*P):Y=96*SIN(AY*P):PSET (X,Y)
580 FOR I=AY TO AZ STEP 2
590 X=96*COS(I*P):Y=96*SIN(I*P)
600 LINE -(X,Y)
610 NEXT
620 D=AY:GOSUB 690
630 D=D+1:GOSUB 690
640 FOR I=1 TO 4
650 D=D+(VP*DP+VN*DN)*AC:GOSUB 690
660 D=D+1:GOSUB 690:NEXT I
670 'LPRINT D
680 RETURN
690 PSET (0,0):DRAW "TA=D; BR101; R14; BL115"
700 RETURN
710 '-----PEN PLOT-----
720 PX=CX+36*COS(AY*P):PY=CY+36*SIN(AY*P)
730 PRINT#1,"PA";PX;PY;"PDAA";CX;CY;AZ-AY;"PU;"
740 PX=CX+33*COS(AY*P):PY=CY+33*SIN(AY*P)
750 PRINT#1,"PA";PX;PY;"PDAA";CX;CY;AZ-AY;"PU;"
760 FOR J=0 TO NM
770 PX=CX+33*COS((AY+((J*(VP*DP+VN*DN+LI))*AC))*P)
780 PY=CY+33*SIN((AY+((J*(VP*DP+VN*DN+LI))*AC))*P)
790 EX=CX+38*COS((AY+((J*(VP*DP+VN*DN+LI))*AC))*P)
800 EY=CY+38*SIN((AY+((J*(VP*DP+VN*DN+LI))*AC))*P):GOSUB 870
810 PX=CX+33*COS((AY+((LI+(J*(VP*DP+VN*DN+LI))*AC))*P)
820 PY=CY+33*SIN((AY+((LI+(J*(VP*DP+VN*DN+LI))*AC))*P)
830 EX=CX+38*COS((AY+((LI+(J*(VP*DP+VN*DN+LI))*AC))*P)
840 EY=CY+38*SIN((AY+((LI+(J*(VP*DP+VN*DN+LI))*AC))*P):GOSUB 870
850 NEXT J
860 RETURN
870 PRINT#1,"PA";PX;PY;"PDPA";EX;EY;"PU;":RETURN

```

4.4 GRAY.BAS Listing

```

10 REM-----GRAY-----
20 COLOR 7,1:SCREEN 0:CLS:HX$="0123456789ABCDEF":LPRINT CHR$(27)"["CHR$(5)
30 PRINT" -- MENU --":PRINT
40 PRINT"1. INPUT GRAY, COMPUTE BINARY AND DECIMAL"
50 PRINT"2. INPUT DECIMAL, COMPUTE BINARY AND GRAY"
60 PRINT"3. INPUT DECIMAL TOP, COMPUTE BINARY AND GRAY"
70 PRINT"4. INPUT 16 BIT GRAY WORD, COMPUTE BINARY AND DECIMAL PARTS"
80 PRINT"5. INPUT POINTING ANGLES, COMPUTE GRAY WORD"
90 PRINT"6. PRINT A TABLE OF VALUES, GRAY CODE, AND ANGLES"
100 PRINT"7. QUIT"
110 PRINT:INPUT"CHOICE ";Z
120 IF Z<1 OR Z>7 THEN 110
130 CLS
140 IF Z=2 THEN 280
150 IF Z=3 THEN 360
160 IF Z=4 THEN 450
170 IF Z=5 THEN 620
180 IF Z=6 THEN 1340
190 IF Z=7 THEN END
200 PRINT:INPUT"INPUT GRAY CODE ";G$
210 GOSUB 740
220 PRINT"BINARY = ";B$
230 GOSUB 840
240 PRINT"DECIMAL = ";D
250 PRINT:PRINT"PRESS ANY KEY TO RETURN TO MENU"
260 Z$=INPUT$(1):IF Z$="" THEN 260
270 CLS:GOTO 30
280 PRINT:INPUT"INPUT DECIMAL NUMBER ";D
290 GOSUB 960

```

```

300 PRINT"BINAR Y   = ";B$
310 GOSUB 1030
320 PRINT"GRAY CODE = ";G$
330 PRINT:PRINT"PRESS ANY KEY TO RETURN TO MENU"
340 Z$=INPUT$(1):IF Z$="" THEN 340
350 CLS:GOTO 30
360 PRINT:INPUT"INPUT TOP NUMBER ";T
370 FOR J=0 TO T:D=J:B$=""
380 GOSUB 960
390 GOSUB 1030
400 PRINT J;TAB(11)B$;TAB(21)G$
410 NEXT J
420 PRINT:PRINT"PRESS ANY KEY TO RETURN TO MENU"
430 Z$=INPUT$(1):IF Z$="" THEN 430
440 CLS:GOTO 30
450 PRINT:INPUT"INPUT 16 BIT GRAY CODE WORD ";W$
460 IF LEN(W$)<>16 THEN 450
470 GX$=LEFT$(W$,8):GY$=RIGHT$(W$,8)
480 G$=GX$:GOSUB 740:BX$=B$:GOSUB 840:DX=D
490 G$=GY$:B$="":GOSUB 740:BY$=B$:D=0:GOSUB 840:DY=D
500 PRINT:PRINT"16 BIT GRAY WORD = ";W$
510 PRINT"X-AXIS BINARY AND DECIMAL = ";BX$;"  ";DX
520 PRINT"Y-AXIS BINARY AND DECIMAL = ";BY$;"  ";DY
530 GOSUB 1120
540 IF ABS(XZ)>64 THEN PRINT"X-AXIS OUT OF RANGE":GOTO 450
550 PRINT"X-AXIS POINTING ANGLE = ";XZ;" DEGREES"
560 IF ABS(YZ)>64 THEN PRINT"Y-AXIS OUT OF RANGE":GOTO 450
570 PRINT"Y-AXIS POINTING ANGLE = ";YZ;" DEGREES"
580 PRINT"SUN POINTING ANGLE = ";SA;" DEGREES"
590 PRINT:PRINT"PRESS ANY KEY TO RETURN TO MENU"
600 Z$=INPUT$(1):IF Z$="" THEN 600
610 CLS:GOTO 30
620 PRINT:INPUT"INPUT X AND Y-AXIS POINTING ANGLES IN DEGREES ";XZ,YZ
630 X1=XZ:Y1=YZ
640 IF XZ=0 THEN XZ=.000001
650 IF YZ=0 THEN YZ=.000001
660 GOSUB 1230:D=INT(DX+.5):GOSUB 960:GOSUB 1030:W$=G$
670 D=INT(DY+.5):B$="":G$="":GOSUB 960:GOSUB 1030:W$=W$+" "+G$
680 PRINT:PRINT"X AND Y-AXIS POINTING ANGLES = ";X1;" AND ";Y1;" DEGREES"
690 GOSUB 1180:PRINT:PRINT"SUN POINTING ANGLE = ";SA;" DEGREES"
700 PRINT:PRINT"GRAY WORD = ";W$
710 PRINT:PRINT"PRESS ANY KEY TO RETURN TO MENU"
720 Z$=INPUT$(1):IF Z$="" THEN 720
730 CLS:GOTO 30
740 REM-----CONVERT GRAY CODE TO BINARY-----
750 Z=LEN(G$)
760 B$=LEFT$(G$,1)
770 FOR I=2 TO Z
780 IF MID$(B$,I-1,1)="1" THEN 800
790 B$=B$+MID$(G$,I,1):GOTO 820
800 IF MID$(G$,I,1)="0" THEN B$=B$+"1":GOTO 820
810 B$=B$+"0"
820 NEXT
830 RETURN
840 REM-----CONVERT BINARY TO DECIMAL-----
850 Z=LEN(B$):FOR I=1 TO Z
860 D=D+VAL(MID$(B$,I,1))*2^(Z-I)
870 NEXT
880 RETURN
890 REM-----CONVERT GRAY CODE TO HEX-----
900 G1=0:G2=0:FOR I=3 TO 0 STEP -1
910 G1=G1+VAL(MID$(G$,8-I,1))*2^I:NEXT
920 FOR I=7 TO 4 STEP -1
930 G2=G2+VAL(MID$(G$,8-I,1))*2^(I-4):NEXT
940 G1$=MID$(HX$,G1+1,1):G2$=MID$(HX$,G2+1,1)
950 GH$=G2$+G1$:RETURN
960 REM-----CONVERT DECIMAL TO BINARY-----
970 B$=""
980 FOR I=7 TO 0 STEP -1
990 Z=D-2^I:IF Z<0 THEN B$=B$+"0":GOTO 1010
1000 D=Z:B$=B$+"1"
1010 NEXT
1020 RETURN

```



```

1030 REM-----CONVERT BINARY TO GRAY CODE-----
1040 G$=LEFT$(B$,1)
1050 FOR I=2 TO 8
1060 IF MID$(B$,I-1,1)="1" THEN 1080
1070 G$=G$+MID$(B$,I,1):GOTO 1100
1080 IF MID$(B$,I,1)="0" THEN G$=G$+"1":GOTO 1100
1090 G$=G$+"0"
1100 NEXT
1110 RETURN
1120 REM-----COMPUTE POINTING ANGLES-----
1130 X=.00275*DX-.350625:Y=.00275*DY-.350625
1140 XY=X^2+Y^2
1150 IF XY>.179536952# THEN PRINT:PRINT"GRAY CODE OUT OF RANGE":END
1160 XT=(1.4553*X)/((.200704-1.117898*(X^2+Y^2))^5)
1170 YT=(1.4553*Y)/((.200704-1.117898*(X^2+Y^2))^5)
1180 XZ=(180*ATN(XT))/3.14159
1190 YZ=(180*ATN(YT))/3.14159
1200 ST=(1.4553*((X^2+Y^2)^5))/((.200704-1.117898*(X^2+Y^2))^5)
1210 SA=(180*ATN(ST))/3.14159
1220 RETURN
1230 REM-----CONVERT ANGLES TO 16 BIT GRAY WORD-----
1240 PI=3.14159
1250 XZ=PI*XZ/180:YZ=PI*YZ/180
1260 K1=1.4553^2:K2=.200704:K3=1.117898:K4=(TAN(XZ)/TAN(YZ))^2:K5=(TAN(XZ))^2
1270 C4=.00275:C5=.350625
1280 Y=((K2*K5)/(K1*K4+K3*K4*K5+K3*K5))^5
1290 IF Y1<0 THEN Y=-Y
1300 X=ABS(Y)*(K4^5)
1310 IF X1<0 THEN X=-X
1320 DX=(X+C5)/C4:DY=(Y+C5)/C4
1330 RETURN
1340 REM-----TABLE/DECIMAL, GRAY, ANGLE-----
1350 LPRINT TAB(23)"TABLE OF GRAY CODE BYTES AND ANGLES":LPRINT:LPRINT
1360 LPRINT TAB(5)"DECIMAL"TAB(18)"BINARY"TAB(30)"HEX"TAB(41)"GRAY CODE"TAB(54)"GRAY HEX"TAB(70)"ANGLE"
1370 LPRINT
1380 CLS
1390 FOR J=0 TO 255:D=J:GOSUB 960:GOSUB 1030:GOSUB 890
1400 X=.00275*J-.350625
1410 XT=(1.4553*X)/((.200704-1.117898*(X^2))^5)
1420 XZ=(180*ATN(XT))/3.14159
1430 J$=HEX$(J):IF LEN(J$)=1 THEN J$="0"+J$
1440 LPRINT TAB(7)J;TAB(17)B$;TAB(31)J$;TAB(41)G$;TAB(58)GH$;
1450 LPRINT TAB(69);:LPRINT USING "###.###";XZ
1460 PRINT J;B$,HEX$(J),G$,GH$,XZ:B$="":G$=""
1470 NEXT
1480 PRINT:PRINT"PRESS ANY KEY TO RETURN TO MENU"
1490 Z$=INPUT$(1):IF Z$="" THEN 720
1500 CLS:GOTO 30
1510 END
1520 SCREEN 0:COLOR 7,1:WIDTH 80:CLS
1530 END

```

TABLE 3.4-1

TABLE OF GRAY CODE BYTES AND ANGLES

0	00000000	0	00000000	00	-63.75868
1	00000001	1	00000001	01	-63.19641
2	00000010	2	00000011	03	-62.63997
3	00000011	3	00000010	02	-62.08894
4	00000100	4	00000110	06	-61.54294
5	00000101	5	00000111	07	-61.00166
6	00000110	6	00000101	05	-60.46475
7	00000111	7	00000100	04	-59.93193
8	00001000	8	00001100	0C	-59.40294
9	00001001	9	00001101	0D	-58.87752
10	00001010	A	00001111	0F	-58.35544
11	00001011	B	00001110	0E	-57.83649
12	00001100	C	00001010	0A	-57.32047
13	00001101	D	00001011	0B	-56.80718
14	00001110	E	00001001	09	-56.29644
15	00001111	F	00001000	08	-55.7881
16	00010000	10	00011000	18	-55.282
17	00010001	11	00011001	19	-54.77798
18	00010010	12	00011011	1B	-54.27593
19	00010011	13	00011010	1A	-53.77569
20	00010100	14	00011110	1E	-53.27715
21	00010101	15	00011111	1F	-52.7802
22	00010110	16	00011101	1D	-52.28472
23	00010111	17	00011100	1C	-51.7906
24	00011000	18	00010100	14	-51.29775
25	00011001	19	00010101	15	-50.80609
26	00011010	1A	00010111	17	-50.31551
27	00011011	1B	00010110	16	-49.82592
28	00011100	1C	00010010	12	-49.33726
29	00011101	1D	00010011	13	-48.84945
30	00011110	1E	00010001	11	-48.3624
31	00011111	1F	00010000	10	-47.87607
32	00100000	20	00110000	30	-47.39035
33	00100001	21	00110001	31	-46.90521
34	00100010	22	00110011	33	-46.42058
35	00100011	23	00110010	32	-45.93641
36	00100100	24	00110110	36	-45.45263
37	00100101	25	00110111	37	-44.9692
38	00100110	26	00110101	35	-44.48606
39	00100111	27	00110100	34	-44.00316
40	00101000	28	00111100	3C	-43.52046
41	00101001	29	00111101	3D	-43.03793
42	00101010	2A	00111111	3F	-42.5555
43	00101011	2B	00111110	3E	-42.07316
44	00101100	2C	00111010	3A	-41.59083
45	00101101	2D	00111011	3B	-41.10852
46	00101110	2E	00111001	39	-40.62617
47	00101111	2F	00111000	38	-40.14375

48	00110000	30	00101000	28	-39.66122
49	00110001	31	00101001	29	-39.17856
50	00110010	32	00101011	2B	-38.69574
51	00110011	33	00101010	2A	-38.21273
52	00110100	34	00101110	2E	-37.7295
53	00110101	35	00101111	2F	-37.24604
54	00110110	36	00101101	2D	-36.7623
55	00110111	37	00101100	2C	-36.27829
56	00111000	38	00100100	24	-35.79395
57	00111001	39	00100101	25	-35.3093
58	00111010	3A	00100111	27	-34.8243
59	00111011	3B	00100110	26	-34.33893
60	00111100	3C	00100010	22	-33.85316
61	00111101	3D	00100011	23	-33.36701
62	00111110	3E	00100001	21	-32.88044
63	00111111	3F	00100000	20	-32.39344
64	01000000	40	01100000	60	-31.906
65	01000001	41	01100001	61	-31.41811
66	01000010	42	01100011	63	-30.92975
67	01000011	43	01100010	62	-30.4409
68	01000100	44	01100110	66	-29.95158
69	01000101	45	01100111	67	-29.46176
70	01000110	46	01100101	65	-28.97144
71	01000111	47	01100100	64	-28.4806
72	01001000	48	01101100	6C	-27.98926
73	01001001	49	01101101	6D	-27.49739
74	01001010	4A	01101111	6F	-27.00499
75	01001011	4B	01101110	6E	-26.51207
76	01001100	4C	01101010	6A	-26.01861
77	01001101	4D	01101011	6B	-25.52461
78	01001110	4E	01101001	69	-25.03008
79	01001111	4F	01101000	68	-24.535
80	01010000	50	01111000	78	-24.03939
81	01010001	51	01111001	79	-23.54324
82	01010010	52	01111011	7B	-23.04655
83	01010011	53	01111010	7A	-22.54932
84	01010100	54	01111110	7E	-22.05155
85	01010101	55	01111111	7F	-21.55326
86	01010110	56	01111101	7D	-21.05443
87	01010111	57	01111100	7C	-20.55508
88	01011000	58	01110100	74	-20.05521
89	01011001	59	01110101	75	-19.55483
90	01011010	5A	01110111	77	-19.05393
91	01011011	5B	01110110	76	-18.55253
92	01011100	5C	01110010	72	-18.05064
93	01011101	5D	01110011	73	-17.54825
94	01011110	5E	01110001	71	-17.04538
95	01011111	5F	01110000	70	-16.54205
96	01100000	60	01010000	50	-16.03824
97	01100001	61	01010001	51	-15.53398
98	01100010	62	01010011	53	-15.02927
99	01100011	63	01010010	52	-14.52413

100	01100100	64	01010110	56	-14.01856
101	01100101	65	01010111	57	-13.51259
102	01100110	66	01010101	55	-13.0062
103	01100111	67	01010100	54	-12.49943
104	01101000	68	01011100	5C	-11.99227
105	01101001	69	01011101	5D	-11.48476
106	01101010	6A	01011111	5F	-10.97688
107	01101011	6B	01011110	5E	-10.46867
108	01101100	6C	01011010	5A	-9.960138
109	01101101	6D	01011011	5B	-9.451286
110	01101110	6E	01011001	59	-8.942136
111	01101111	6F	01011000	58	-8.43271
112	01110000	70	01001000	48	-7.923009
113	01110001	71	01001001	49	-7.413052
114	01110010	72	01001011	4B	-6.902863
115	01110011	73	01001010	4A	-6.392446
116	01110100	74	01001110	4E	-5.881821
117	01110101	75	01001111	4F	-5.371009
118	01110110	76	01001101	4D	-4.860017
119	01110111	77	01001100	4C	-4.348866
120	01111000	78	01000100	44	-3.837576
121	01111001	79	01000101	45	-3.326155
122	01111010	7A	01000111	47	-2.814624
123	01111011	7B	01000110	46	-2.303006
124	01111100	7C	01000010	42	-1.791305
125	01111101	7D	01000011	43	-1.279545
126	01111110	7E	01000001	41	-.7677432
127	01111111	7F	01000000	40	-.2559209
128	10000000	80	11000000	C0	.2559153
129	10000001	81	11000001	C1	.7677432
130	10000010	82	11000011	C3	1.27954
131	10000011	83	11000010	C2	1.7913
132	10000100	84	11000110	C6	2.303
133	10000101	85	11000111	C7	2.814618
134	10000110	86	11000101	C5	3.32615
135	10000111	87	11000100	C4	3.837571
136	10001000	88	11001100	CC	4.348866
137	10001001	89	11001101	CD	4.860011
138	10001010	8A	11001111	CF	5.371003
139	10001011	8B	11001110	CE	5.881821
140	10001100	8C	11001010	CA	6.39244
141	10001101	8D	11001011	CB	6.902857
142	10001110	8E	11001001	C9	7.413052
143	10001111	8F	11001000	C8	7.923003
144	10010000	90	11011000	D8	8.432704
145	10010001	91	11011001	D9	8.942136
146	10010010	92	11011011	DB	9.451281
147	10010011	93	11011010	DA	9.960131
148	10010100	94	11011110	DE	10.46867
149	10010101	95	11011111	DF	10.97688
150	10010110	96	11011101	DD	11.48475
151	10010111	97	11011100	DC	11.99227

152	10011000	98	11010100	D4	12.49942
153	10011001	99	11010101	D5	13.0062
154	10011010	9A	11010111	D7	13.51258
155	10011011	9B	11010110	D6	14.01856
156	10011100	9C	11010010	D2	14.52412
157	10011101	9D	11010011	D3	15.02927
158	10011110	9E	11010001	D1	15.53398
159	10011111	9F	11010000	D0	16.03823
160	10100000	A0	11110000	F0	16.54204
161	10100001	A1	11110001	F1	17.04538
162	10100010	A2	11110011	F3	17.54825
163	10100011	A3	11110010	F2	18.05063
164	10100100	A4	11110110	F6	18.55253
165	10100101	A5	11110111	F7	19.05393
166	10100110	A6	11110101	F5	19.55482
167	10100111	A7	11110100	F4	20.05521
168	10101000	A8	11111100	FC	20.55508
169	10101001	A9	11111101	FD	21.05443
170	10101010	AA	11111111	FF	21.55325
171	10101011	AB	11111110	FE	22.05155
172	10101100	AC	11111010	FA	22.54931
173	10101101	AD	11111011	FB	23.04654
174	10101110	AE	11111001	F9	23.54324
175	10101111	AF	11111000	F8	24.03938
176	10110000	B0	11101000	E8	24.535
177	10110001	B1	11101001	E9	25.03007
178	10110010	B2	11101011	EB	25.52461
179	10110011	B3	11101010	EA	26.0186
180	10110100	B4	11101110	EE	26.51207
181	10110101	B5	11101111	EF	27.00499
182	10110110	B6	11101101	ED	27.49738
183	10110111	B7	11101100	EC	27.98926
184	10111000	B8	11100100	E4	28.4806
185	10111001	B9	11100101	E5	28.97143
186	10111010	BA	11100111	E7	29.46175
187	10111011	BB	11100110	E6	29.95157
188	10111100	BC	11100010	E2	30.4409
189	10111101	BD	11100011	E3	30.92974
190	10111110	BE	11100001	E1	31.4181
191	10111111	BF	11100000	E0	31.906
192	11000000	C0	10100000	A0	32.39344
193	11000001	C1	10100001	A1	32.88043
194	11000010	C2	10100011	A3	33.36701
195	11000011	C3	10100010	A2	33.85316
196	11000100	C4	10100110	A6	34.33891
197	11000101	C5	10100111	A7	34.8243
198	11000110	C6	10100101	A5	35.3093
199	11000111	C7	10100100	A4	35.79395
200	11001000	C8	10101100	AC	36.27829
201	11001001	C9	10101101	AD	36.7623
202	11001010	CA	10101111	AF	37.24603
203	11001011	CB	10101110	AE	37.7295

204	11001100	CC	10101010	AA	38.21272
205	11001101	CD	10101011	AB	38.69573
206	11001110	CE	10101001	A9	39.17855
207	11001111	CF	10101000	A8	39.66121
208	11010000	D0	10111000	B8	40.14375
209	11010001	D1	10111001	B9	40.62617
210	11010010	D2	10111011	BB	41.10851
211	11010011	D3	10111010	BA	41.59083
212	11010100	D4	10111110	BE	42.07316
213	11010101	D5	10111111	BF	42.5555
214	11010110	D6	10111101	BD	43.03793
215	11010111	D7	10111100	BC	43.52046
216	11011000	D8	10110100	B4	44.00315
217	11011001	D9	10110101	B5	44.48606
218	11011010	DA	10110111	B7	44.9692
219	11011011	DB	10110110	B6	45.45262
220	11011100	DC	10110010	B2	45.93641
221	11011101	DD	10110011	B3	46.42058
222	11011110	DE	10110001	B1	46.9052
223	11011111	DF	10110000	B0	47.39035
224	11100000	E0	10010000	90	47.87606
225	11100001	E1	10010001	91	48.3624
226	11100010	E2	10010011	93	48.84945
227	11100011	E3	10010010	92	49.33726
228	11100100	E4	10010110	96	49.82592
229	11100101	E5	10010111	97	50.31551
230	11100110	E6	10010101	95	50.80609
231	11100111	E7	10010100	94	51.29775
232	11101000	E8	10011100	9C	51.7906
233	11101001	E9	10011101	9D	52.28471
234	11101010	EA	10011111	9F	52.78019
235	11101011	EB	10011110	9E	53.27715
236	11101100	EC	10011010	9A	53.77569
237	11101101	ED	10011011	9B	54.27592
238	11101110	EE	10011001	99	54.77798
239	11101111	EF	10011000	98	55.282
240	11110000	F0	10001000	88	55.7881
241	11110001	F1	10001001	89	56.29644
242	11110010	F2	10001011	8B	56.80717
243	11110011	F3	10001010	8A	57.32046
244	11110100	F4	10001110	8E	57.83649
245	11110101	F5	10001111	8F	58.35544
246	11110110	F6	10001101	8D	58.87751
247	11110111	F7	10001100	8C	59.40294
248	11111000	F8	10000100	84	59.93192
249	11111001	F9	10000101	85	60.46474
250	11111010	FA	10000111	87	61.00165
251	11111011	FB	10000110	86	61.54294
252	11111100	FC	10000010	82	62.08894
253	11111101	FD	10000011	83	62.63996
254	11111110	FE	10000001	81	63.19641
255	11111111	FF	10000000	80	63.75868

APPENDIX G - PCB POWER CONNECTIONS

Note: Power supply boards' output jacks are listed. All other equipment or boards' input plugs are listed.

PCB Name	Plug #	Pin #	Power Conn.	Init & Date
AD1	P1	1	+5VDAC	
		2	+5VDAC	
		3	+5VDAC	
		4	RTNDAC	
		5	RTNDAC	
		6	RTNDAC	
		31	RTNF	
		32	-15VF	
		36	+5VDAC	
		37	+5VDAC	
		38	+5VDAC	
		39	RTNDAC	
		40	RTNDAC	
		41	RTNDAC	
		66	RTNF	
		67	+15VF	
AD2	P1	1	+5VDAC	
		2	+5VDAC	
		3	+5VDAC	
		4	RTNDAC	
		5	RTNDAC	
		6	RTNDAC	
		31	RTNF	
		32	-15VF	
		36	+5VDAC	
		37	+5VDAC	
		38	+5VDAC	
		39	RTNDAC	
		40	RTNDAC	
		41	RTNDAC	
		66	RTNF	
		67	+15VF	
APS	J1	1	+28V SW IN	
		2	+28V SW IN	
		3	+28V SW RTN	
		4	+28V SW RTN	
		5	+28V SW HV RTN	
		6	+28V SW HV RTN	
		7	+28V SW HV OUT	
		8	+28V SW HV OUT	
		9	+28V SW PS1 OUT	
		11	+28V SW PS1 RTN	
		13	+28V SW HV RTN	
		15	+28V SW HV OUT	

J2	1	+5V EMITTER
	2	+5V EMITTER
	3	+5V EMITTER
	4	+5V EMITTER
	6	EMITTER RTN
	7	EMITTER RTN
	8	EMITTER RTN
	9	EMITTER RTN
J3	1	+5V ASIS
	3	+15V ASIS
	5	-15V ASIS
	7	+15V AD210
	9	+24V RLY1
	11	+24V RLY2
	13	+15V HVE
	15	-15V HVE
	20	+5V ASIS RTN
	22	+15V ASIS RTN
	24	-15V ASIS RTN
	26	+15V AD210 RTN
	28	+24V RLY1 RTN
	30	+24V RLY2 RTN
	32	+/-15V HVE RTN
J4	2	+5V EMITTER
	4	+5V ASIS
	6	+15V ASIS
	8	-15V ASIS
	10	+24V RLY
	12	+15V AD210
	14	-15V HVE
	16	+15V HVE
	21	EMITTER RTN
	23	+5V ASIS RTN
	25	+/-15V ASIS RTN
	27	+/-15V ASIS RTN
	29	+24V RLY RTN
	31	+15V AD210 RTN
	33	+/-15V HVE RTN
	35	+/-15V HVE RTN
CMD P1	1	+5VDAC
	2	+5VDAC
	3	+5VDAC
	4	RTNDAC
	5	RTNDAC
	6	RTNDAC
	7	+5VLO
	36	+5VDAC
	37	+5VDAC
	38	+5VDAC
	39	RTNDAC
	40	RTNDAC
	41	RTNDAC

	P2	42	+5VLO
		4	+5VLO
		5	+5VLO
		6	+5VLO
		8	+5VMO
		9	CMD RTN
		10	CMD RTN
		13	RTNLO
		15	RTNMO
DBP	P1	1	+5V DAC SENSE
		2	+5VDAC
		3	+5VDAC
		5	-15V DAC
		7	+15V DAC
		9	RTNDAC
		10	RTNDAC
		13	-15V RTN
		15	+15V RTN
DIG	P1	1	+5VDAC
		2	+5VDAC
		3	+5VDAC
		4	RTNDAC
		5	RTNDAC
		6	RTNDAC
		36	+5VDAC
		37	+5VDAC
		38	+5VDAC
		39	RTNDAC
		40	RTNDAC
		41	RTNDAC
	P2	9	+5VDAC
		46	RTNDAC
		47	RTNDAC
		48	RTNDAC
		49	+5VDAC
		50	+5VDAC
	P3	8	RTNDAC
		9	RTNDAC
HK1	P1	6	+15VDAC
		7	-15VDAC
		15	GND
		16	GND
		18	RTNDAC
		19	RTNDAC
	P2	2	+5V EMITTER
		4	+5V ASIS
		6	+15V ASIS
		8	-15V ASIS
		10	+24V RLY
		12	+15V AD210
		14	-15V HVE
		16	+15V HVE

		21	EMITTER RTN
		23	+5V ASIS RTN
		25	+15V ASIS RTN
		27	-15V ASIS RTN
		29	+24V RLY RTN
		31	+15V AD210 RTN
		33	-15V HVE RTN
		35	+15V HVE RTN
HK2	P1	6	+15VDAC
		7	-15VDAC
		15	GND
		16	GND
		18	RTNDAC
		19	RTNDAC
	P2	2	+5V AK
		4	+5V MO
		6	+5V LO
		8	+5V DAC
		10	+15V DAC
		12	-15V DAC
		14	+28V SW
		16	+5V SPARE
		21	+5V AK RTN
		23	+5V MO RTN
		25	+5V LO RTN
		27	+5V DAC RTN
		29	+15V DAC RTN
		31	-15V DAC RTN
		33	+28V SW RTN
		35	+5V SPARE RTN
HVC	P1	1	+5VDAC
		16	DACS RTN
	P2	1	+15V AD210
		2	-15V ASIS
		4	+15V ASIS
		5	+5V ASIS
		10	+15V HVE RTN
		14	+15V AD210 RTN
		15	+15V AD210
		16	+15V AD210
		21	+/-15V & +5V ASIS RTN
		23	+15V HVE
		25	-15V HVE
HVD	P1	1	+15V AD210 RTN
		9	+15V AD210
		10	+15V REG RTN
		13	+15V REG IN
	J2	1	+28V RTN
		2	+28V RTN
		4	+2-25V OUT
		5	+2-25V OUT
		6	+28V RTN

		9	+2-25V OUT
	P3	1	+28V RTN
		2	+28V RTN
		4	+28V IN
		5	+28V IN
		6	+28V RTN
		9	+28V IN
HVE	P1	5	+24V RLY1
		11	+15V SW AD210
		12	S/C GND
		18	-15V HVE
		19	+15V HVE
		22	+24V RLY1
		23	+/-15V HVE RTN
		25	+/-15V AD210 & +24V RLY RTN .
	P2	1	HV RTN
		5	HV IN
		7	HV OUTPUT
HVO	P1	3	+15V REG OUT
		8	+15V SW AD210
		12	+15V AD210
		14	+24V RLY1
		15	+15V REG RTN
	P2	1	+28V RTN
		5	+28V IN
	J3	1	HV OUT
		5	HV RTN
	P4	1	+28V RTN
		2	+28V RTN
		4	+2-25V IN
		5	+2-25V IN
		6	+28V RTN
		9	+2-25V IN
IVN	P1	1	+5V ASIS
		7	+24V RLY2
		8	+24V RLY1&2 RTN
		14	+24V RLY1
		15	+15V AD210
LPE	P1	1	+28V S/C
		2	+28V S/C
		6	+28V S/C RTN
		7	+28V S/C RTN
	P2	7	+15V
		9	+5V DACS
		19	+15V RTN
MET	P1	1	+5VDAC
		2	+5VDAC
		3	+5VDAC
		4	RTNDAC
		5	RTNDAC
		6	RTNDAC
		36	+5VDAC

		37	+5VDAC
		38	+5VDAC
		39	RTNDAC
		40	RTNDAC
		41	RTNDAC
	P2	8	+5V AK
		9	RTNAK
		15	RTNAK
PGE	P1	A	+28V S/C
		E	+28V RTN
PS1	P1	A	+28V SW
		D	+28V RTN
PS2	P1	A	+28V SW
		D	+28V RTN
RDO	P1	1	+5VDAC
		2	+5VDAC
		3	+5VDAC
		4	RTNDAC
		5	RTNDAC
		6	RTNDAC
		36	+5VDAC
		37	+5VDAC
		38	+5VDAC
		39	RTNDAC
		40	RTNDAC
		41	RTNDAC
	P2	8	+5V MO
		11	RTNMO
		12	RTNMO
		15	RTNMO
RLL	P1	1	+5VDAC
		13	+5VDAC
	P3	3	+15V SW AD210 RTN
		4	+15V SW AD210
		7	+5V ASIS RTN
		8	+5V ASIS
		13	S/C GND
		20	+24V RLY1 RTN
		25	+24V RLY1
	P4	1	+5V EMITTER OUT
		2	+5V EMITTER OUT
		4	+5V EMITTER IN
		5	+5V EMITTER IN
		6	+5V EMITTER OUT
		7	+5V EMITTER OUT
		8	+5V EMITTER IN
		9	+5V EMITTER IN
RLR	P1	1	+24V RLY1
		5	+24V RLY2
		13	S/C GND
		14	S/C GND
		18	+24V RLY RTN

RTD	P4	5	HV+/-	
	P1	1	+5VDAC	
		2	+5VDAC	
		3	+5VDAC	
		4	RTNDAC	
		5	RTNDAC	
		6	RTNDAC	
		31	RTNF	
		32	-15VF	
		36	+5VDAC	
		37	+5VDAC	
		38	+5VDAC	
		39	RTNDAC	
		40	RTNDAC	
		41	RTNDAC	
		66	RTNF	
		67	+15VF	
	SEQ	P1	1	+5VDAC
			2	+5VDAC
			3	+5VDAC
		4	RTNDAC	
		5	RTNDAC	
		6	RTNDAC	
		36	+5VDAC	
		37	+5VDAC	
		38	+5VDAC	
		39	RTNDAC	
		40	RTNDAC	
		41	RTNDAC	
SPS	P1	A	+28V IN	
		B	+28V RTN	
		C	+28V RTN	
		D	+28V IN	
	J2	A	+28V SW OUT	
		B	+28V SW OUT RTN	
		C	+28V SW OUT RTN	
		D	+28V SW OUT	
	J3	1	+5V MO CMD	
		3	+5V MO RDO	
		7	+5V AK SSI	
		9	+5V AK TDO	
		11	+5V AK MET	
		14	RTN MO CMD	
		16	RTN MO RDO	
		20	RTN AK SSI	
		22	RTN AK TDO	
		24	RTN AK MET	
		J4	1	+5V LO CMD
			2	+5V LO SSI
5	+5V DBP			
6	+5V DACS LPE			
7	+5V DACS HK1			

	8	+5V DACS HK2
	9	+5V DACS HVC
	10	+5V DACS RLL
	11	+5V DBP
	12	+15V DACS DBP
	13	+15V DACS HK1
	14	+15V DACS HK2
	15	+15V DACS LPE
	17	-15V DACS DBP
	18	-15V DACS HK1
	19	-15V DACS HK2
	20	+5V RTN LO CMD
	21	+5V RTN LO SSI
	23	+5V RTN DAC DBP
	24	+5V RTN DAC DBP
	26	+5V RTN DAC HK1
	27	+5V RTN DAC HK2
	28	+5V RTN DAC HVC
	30	+15V RTN DBP
	31	+15V RTN HK1
	32	+15V RTN HK2
	33	+15V RTN LPE
	35	-15V RTN DBP
	36	-15V RTN HK1
	37	-15V RTN HK2
J5	2	+5V AK
	4	+5V MO
	6	+5V LO
	8	+5V DACS
	10	+15V DACS
	12	-15V DACS
	14	+28V SW
	21	+5V AK RTN
	23	+5V MO RTN
	25	+5V LO RTN
	27	+5V DACS RTN
	29	+15V DACS
	31	-15V DACS
	33	+28V SW RTN
J6	1	+28V SW PS2
	2	+28V SW PGE
	3	+28V SW LPE
	4	+28V SW TPM
	7	+28V SSE
	13	+5VDAC SPS
	14	+28V SW PS2 RTN
	15	+28V SW PGE RTN
	16	+28V SW LPE RTN
	17	+28V SW TPM RTN
	20	+28V SSE RTN
	22	S/C GND
	23	S/C GND

G-9

		41	RTNDAC
TPM	P1	1	+28V
		2	+28V RTN

APPENDIX H - FLIGHT SOFTWARE LISTING

The PASP flight software listing consists of 43 pages of FORTH code screens (0-127 & 130) on the left side of the page and the attendant shadow screens (150-277 & 280) on the right side of the page. Each screen contains 16 lines for code with each line containing up to 64 characters. The first three spaces of each line are used for line number display and are not counted as part of the line characters.

Shadow screens provide a self documenting feature to the software so there will be no screen by screen detailed commentary on the software. Discovering the general context of each screen is not difficult but a detailed understanding of the flight coding will be eased by first looking at the data dictionary located in Appendix L.

Flight software is written in the F83 version of FORTH which is based on the FORTH-83 Standard (Reference: FORTH-83 Standard, A Publication of the FORTH Standards Team, Mountain View Press, August 1983). The Metacompiler that was used is found in the meta.blk program on the GSE controller hard disk.

After compilation, the memory map (See Appendix I) shows that 1C50 (HEX) bytes of ROM memory is used out of the available 8K. This translates into 944 bytes of unused memory out of 8192 total bytes for a margin of 11.5%.

GSE Menu Software Listing

Page# 1

C:PASP.BLK

0
0 \ Screen # 67
5JUL90CDB
1 \ Screen # 124
2 \ Screen # 125
3 \ Screen # 147
4 \ Screen # 147
5 \ Screen # 148
6 \ Screen # 132
7 \ Screen # 132
8 \ Screen # 124
9 \ Screen # 93
10 \ Screen # 148
11 \ Screen # 148
12 \ Screen # 148
13 \ Screen # 102
14 \ Screen # 148
15 \ Screen # 148

1

0 \ DACS SOFTWARE LOAD SCREEN
15FEB90CDB
1 DECIMAL
2 \ 6 123 THRU 125 126 THRU
3 6 123 T-THRU 125 126 T-THRU \ DOWN LOADER
4 CR .(Dacs.blk is Loaded) HERE H.

5 DECIMAL
6
7
8
9
10
11
12
13
14
15

2

0 \ DACS METACOMPILIER LOAD SCREEN
15FEB90CDB
1 HEX
2
3 DECIMAL
4 29 126 THRU \ DOWN LOADER
5 CR .(Dacs.blk is Loaded) HERE H.
6 DECIMAL
7
8
9

10
11
12
13
14
15

150

18JUL90CDB \

15JUL90CDB
15JUL90CDB
5JUL90CDB
5JUL90CDB
5JUL90CDB
19JUN90CDB
19JUN90CDB
19JUN90CDB
16MAY90CDB
10MAY90CDB
10MAY90CDB
10MAY90CDB
10MAY90CDB
9MAY90CDB
9MAY90CDB

151

3APR90CDB \ DACS SOFTWARE LOAD SCREEN

DECIMAL

This screen is used to down load the flight software for system development and debugging .
The Dacs must contain the development version of the SEQ board

152

14JUL90CDB \ DACS METACOMPILIER LOAD SCREEN

DECIMAL

Called from METAFULL.BLK (SCR#1)
This screen is used to down load the flight software for Metacompiling .
A system rom-image is compiled at E000-FFFF. The D-GSE.COM program will allow the saving to disk of the rom-image .
(SAVE-ROMS)

The Dacs must contain the development version of the SEQ board

Forth 83 Model

<p>3</p> <p>0 \ DACS METACOMPILIER LOAD SCREEN - Dev memory map</p> <p>15FEB90CDB</p> <p>1 HEX</p> <p>2 0000 CONSTANT C-FIELD</p> <p>)</p> <p>3 E000 CONSTANT ROM-START</p> <p>4 ROM-START CONSTANT CODE-START</p> <p>)</p> <p>5 FFF0 CONSTANT RESET-VEC</p> <p>6 \ ROM-START 2000 ERASE</p> <p>7 3000 CONSTANT RAM-SIZE (4000 in flight system 1000 PMEM)</p> <p>8 0000 CONSTANT RAM-START (0000 in flight system)</p> <p>9 0100 CONSTANT VEC-FIELD</p> <p>10 0200 CONSTANT R-SIZE</p> <p>11 8000 CONSTANT DATA-START</p> <p>12 DATA-START 1000 + CONSTANT INIT-RO (INIT R-STACK)</p> <p>13 INIT-RO R-SIZE - CONSTANT INIT-SO (INIT S-STACK)</p> <p>14 DECIMAL</p> <p>15</p>	<p>153</p> <p>7JUL90CDB \ DACS METACOMPILIER LOAD SCREEN - Rom memory map</p> <p>HEX</p> <p>C-FIELD - CS register contents (0000 for RAM, F000 for ROM</p> <p>ROM-START - Beginning address of the ROM (Hardware select</p> <p>CODE-START - Start of compiled code.</p> <p>RESET-VEC - Location of 8086 reset vector.</p> <p>ROM-START 2000 ERASE - Zero out the rom-image area.</p> <p>DECIMAL</p>
---	--

<p>4</p> <p>0 \ DACS METACOMPILIER LOAD SCREEN - Flight memory map</p> <p>15FEB90CDB</p> <p>1 HEX</p> <p>2 0000 CONSTANT C-FIELD</p> <p>3 E000 CONSTANT ROM-START</p> <p>4 ROM-START CONSTANT CODE-START</p> <p>5 FFF0 CONSTANT RESET-VEC</p> <p>6 \ ROM-START 2000 ERASE</p> <p>7 4000 CONSTANT RAM-SIZE (4000 in flight system 1000 PMEM)</p> <p>8 0000 CONSTANT RAM-START (0000 in flight system)</p> <p>9 0400 CONSTANT VEC-FIELD</p> <p>10 0800 CONSTANT R-SIZE</p> <p>11 RAM-START VEC-FIELD + CONSTANT DATA-START</p> <p>12 RAM-SIZE RAM-START + 1000 - CONSTANT INIT-RO (INIT R-STACK)</p> <p>13 INIT-RO R-SIZE - CONSTANT INIT-SO (INIT S-STACK)</p> <p>14 DECIMAL</p> <p>15</p>	<p>154</p> <p>7JUL90CDB \ DACS METACOMPILIER LOAD SCREEN - Ram memory map</p> <p>HEX</p> <p>(\ Development rom version of meta compilations)</p> <p>RAM-SIZE - Total ram available.</p> <p>RAM-START - Start of ram (Hardware select)</p> <p>VEC-FIELD - Size of interrupt vector field .</p> <p>DATA-START - Start of usable ram .</p> <p>R-SIZE - Size of forth return stack .</p> <p>INIT-RO - Initial forth return stack.</p> <p>INIT-SO - Initial forth data stack.</p>
--	--

<p>5</p> <p>0 \ DACS METACOMPILIER LOAD SCREEN - System constants</p> <p>15FEB90CDB</p> <p>1</p> <p>2 DECIMAL</p> <p>3</p> <p>4 07 28 THRU</p> <p>5</p> <p>6</p> <p>7</p> <p>8</p> <p>9</p> <p>10</p> <p>11</p> <p>12</p> <p>13</p> <p>14</p> <p>15</p>	<p>155</p> <p>3APR90CDB \ DACS METACOMPILIER LOAD SCREEN - System constants</p> <p>DECIMAL</p> <p>07 31 THRU</p>
---	--

Forth 83 Model

```

6
0 \ DACS SOFTWARE LOAD SCREEN - Hodge Podge
15FEB90CDB
1 HEX
2 warning off empty
the
3 : THERE NOOP ;
4 : ALLOT-R ALLOT ;
5 : IV 21 0 DO C, LOOP ;
6 : DOFILL (S up,lo -- ) DO 90 C, LOOP ;
7 : T: ; ; : T; [ ' ; ] LITERAL EXECUTE ; IMMEDIATE
8 : [TARGET] NOOP ;
9 : INIT-RO RPO @ ; : INIT-SO SPO @ ;
10 0000 CONSTANT C-FIELD
11 E000 CONSTANT RAM-START
12 E020 CONSTANT DATA-START
13 2000 CONSTANT RAM-SIZE
14 DECIMAL
15

```

```

7
0 \ DACS BUS ADDRESSES
15FEB90CDB
1 HEX
2
3 00800 CONSTANT EXT-MEM-BASE
4 EXT-MEM-BASE 0400 + CONSTANT EXT-I/O-BASE
5 EXT-I/O-BASE 0 4 * + CONSTANT SEQ
6 EXT-I/O-BASE 1 4 * + CONSTANT DIG
DIG )
7 EXT-I/O-BASE 2 4 * + CONSTANT PM
8 EXT-I/O-BASE 3 4 * + CONSTANT AD1
9 EXT-I/O-BASE 4 4 * + CONSTANT AD2
10 EXT-I/O-BASE 5 4 * + CONSTANT RTD
board.
11 EXT-I/O-BASE 6 4 * + CONSTANT RDO
12 EXT-I/O-BASE 7 4 * + CONSTANT TDO
13 DECIMAL
14
15

```

```

8
0 \ DACS BUS ADDRESSES - cont'
15FEB90CDB
1 HEX
2 EXT-I/O-BASE 8 4 * + CONSTANT CMD
3 EXT-I/O-BASE 9 4 * + CONSTANT SSI
4 EXT-I/O-BASE 0A 4 * + CONSTANT MET
5 EXT-I/O-BASE 0B 4 * + CONSTANT TPI
6 EXT-I/O-BASE 0C 4 * + CONSTANT TPI-R
7 EXT-I/O-BASE 0D 4 * + CONSTANT SPARE1
8 EXT-I/O-BASE 0E 4 * + CONSTANT SPARE2
9 EXT-I/O-BASE 0F 4 * + CONSTANT COMM
10 EXT-MEM-BASE 000 + CONSTANT RDO-RAM
11 EXT-MEM-BASE 200 + CONSTANT TDO-RAM
12
13 DECIMAL
14
15

```

```

156
3APR90CDB \ DACS SOFTWARE LOAD SCREEN - Hodge Podge

```

```

HEX
These are required to make downloaded code compatible with
Metacompiler code.

```

```

157
5JUL90CDB \ DACS BUS ADDRESSES

```

```

HEX
EXT-MEM-BASE - Base address for Telemetry buffers (RDO,TDO)
EXT-I/O-BASE - Base address for I/O boards.
SEQ - Offset address for Sequence control board.
PM - Offset address for Power monitor ports ( Located on
DIG - Offset address for Digital interface board.
AD1 - Offset address for A/D board # 1 .
AD2 - Offset address for A/D board # 2 .
RTD - Offset address for Resistance Temperature Detector
RDO - Offset address for Real-time Data Output board.
TDO - Offset address for Tape-recorder Data Output board.
DECIMAL

```

```

158
3APR90CDB \ DACS BUS ADDRESSES - cont'

```

```

HEX
CMD - Offset address for Command decoder board.
SSI - Offset address for Sun Sensor Interface board.
MET - Offset address for Mission Elapsed Time board.
TPI - Offset address for Transient Pulse Monitor board.
TPI-R - Offset address for Transient Pulse Monitor Ram.
SPARE1 - Offset address for Spare # 1 board.
SPARE2 - Offset address for Spare # 2 board.
COMM - Offset address for development Communications board.
RDO-RAM - Offset address for Real-time Data Output Ram.
TDO-RAM - Offset address for Tape-recorder Data Output Ram.
DECIMAL

```

Forth 83 Model

```

9
0 \ PRIMARY HEADER - PACKET ID and misc. constants      159
15FEB90CDB \ PRIMARY HEADER - PACKET ID and misc. constants
1 HEX                                                    HEX
2                                                    Telemetry header constants
3 0E8E          CONSTANT   PACKET-ID      C000          CONSTANT   PACKET-CNT   PACKET-ID   PACKET-CNT   P-LENGTH   DEST-RDO   DEST-TDO
STATUS-CNT
4 00FD          CONSTANT   P-LENGTH      AAAA          CONSTANT   DEST-RDO
5 5555          CONSTANT   DEST-TDO      6A04          CONSTANT   STATUS-CNT   FRAME-WORDS - Size of telemetry frame in words ( 16 bit words )
6
7 100 CONSTANT FRAME-WORDS      OF CONSTANT OVERHEAD-WORDS      OVERHEAD-WORDS - Total " NON DATA" words
8 FRAME-WORDS OVERHEAD-WORDS - 2 * CONSTANT DATA-BYTES      DATA-BYTES - Total " DATA " bytes.
9 DATA-BYTES 6 * CONSTANT BUF-SIZE      BUF-SIZE - Data buffer size.
10 544D CONSTANT B-MT          20 CONSTANT E-BUF-SIZE      E-BUF-SIZE - Error buffer size .
11 B-MT - Ascii "MT". Used to initialize data and error buffers.
12 SSI 0 + CONSTANT SSA          SSI 1 + CONSTANT SSX      SSA - Address of Sun Sensor ATA bit.
13 SSI 2 + CONSTANT SSY          SSX - Address of Sun Sensor "X" data.
14 DECIMAL          SSY - Address of Sun Sensor "Y" data.
15

```

```

10
0 \ Protected Memory Support      160
15FEB90CDB \ Protected Memory Support
1 HEX
2
3 40 CONSTANT R-MAX
4 1000 CONSTANT P-SIZE
5 RAM-START 1000 + CONSTANT P-MEM
6 P-MEM CONSTANT R-IMAGE
7 R-IMAGE R-MAX + 4 + CONSTANT P-START
8 P-MEM P-SIZE + CONSTANT P-END
9
10
11
12
13
14
15

```

```

11
0 \ Dacs misc constant and variable declarations      161
15FEB90CDB \ Dacs misc constant and variable declarations
1 HEX
2
3 SEQ 2 + CONSTANT TMR-CLR   SEQ 0 + CONSTANT MET-CLR
4 SEQ 2 + CONSTANT TMR-FLG   SEQ 0 + CONSTANT MET-FLG
interrupt.
5
6 152A CONSTANT BONE   08 CONSTANT OC-BIT
7
8 0 TPI + CONSTANT TPM-CMD   3 TPI + CONSTANT TPM-FMB
9 2 TPI + CONSTANT TPM-ADDR
10 0 TPI + CONSTANT TPM-CDR   2 TPI + CONSTANT TPM-DATA
11 3 TPI + CONSTANT TPM-STAT   5 TPI + CONSTANT TPM-RESET
12
13 DECIMAL
14
15

```

```

TMR-CLR - A write to this address will clear the P-MEM signal.
MET-CLR - A write to this address will generate -INTAK
and clear the MET-INTRQ signal.
TMR-FLG - Reading this address will indicate 2.5 MS.

MET-FLG - Reading this address will indicate MET-INTRQ.
BONE - Code required to satisfy the watch-dog timer.
OC-BIT - Bit mask for the Over Current indication.
TPM-CMD - Transient Pulse Monitor Command port.
TPM-FMB - Transient Pulse Monitor Feed Me Baby port.
TPM-ADDR - Transient Pulse Monitor Address port.
TPM-CDR - Transient Pulse Monitor Control port.
TPM-DATA - Transient Pulse Monitor Data port.
TPM-STAT - Transient Pulse Monitor Status port.
TPM-RESET - Transient Pulse Monitor Reset.

```

Forth 83 Model

12

0 \ DACS Analog Board Support - AD #1 assignments
15FEB90CDB

```

1
2
3 0  CONSTANT PG-PRESS      1  CONSTANT ELECT-I
4 2  CONSTANT ETELEM        3  CONSTANT PRESS2
5 4  CONSTANT IV4-V         5  CONSTANT IV5-V
6 6  CONSTANT IV6-V         7  CONSTANT CAL1A
7
8 0  CONSTANT PG-TEMP       1  CONSTANT EMIT-I
9 2  CONSTANT ITELEM        3  CONSTANT TEMP2-1
10 4  CONSTANT IV4-I        5  CONSTANT IV5-I
11 6  CONSTANT IV6-I        7  CONSTANT CAL1B
12
13
14
15

```

162

11FEB90CDB \ DACS Analog Board Support - AD #1 assignments

```

PG-PRESS - Pressure Gauge Pressure.
ELECT-I - Electrometer current (LogI).
ETELEM - High voltage "VOLTAGE" telemetry.
PRESS2 - Pasp box2 pressure sensor.
IV4-V,IV5-V,IV6-V - IV #s 4,5,6 Voltage channels.
CAL1A, CAL1B - A/D #1 MUX A,B Calibration channels.
PG-TEMP - Pressure Gauge Temperature.
EMIT-I - Emitter emission current channel.
ITELEM - High voltage "CURRENT" telemetry.
TEMP2-1 - Pasp box2 temperature sensor #1.
IV4-I,IV5-I,IV6-I - IV #s 4,5,6 Current channels.

```

13

0 \ DACS Analog Board Support - AD #2 assignments
15FEB90CDB

```

1
2
3 0  CONSTANT TEMP1-2      1  CONSTANT AD2-1
4 2  CONSTANT LP-SWEEP     3  CONSTANT PRESS1
5 4  CONSTANT IV1-V        5  CONSTANT IV2-V
6 6  CONSTANT IV3-V        7  CONSTANT CAL2A
7
8 0  CONSTANT TEMP2-2      1  CONSTANT LP-TEMP
9 2  CONSTANT LP-NE        3  CONSTANT TEMP1-1
10 4  CONSTANT IV1-I       5  CONSTANT IV2-I
11 6  CONSTANT IV3-I       7  CONSTANT CAL2B
12
13 6  CONSTANT RTD-CAL1    7  CONSTANT RTD-CAL2
14 DECIMAL
15

```

163

11FEB90CDB \ DACS Analog Board Support - AD #2 assignments

```

TEMP1-2 - Pasp box1 temperature sensor #2.
AD2-1 - A/D #1 spare channel.
LP-SWEEP - Langmuir Probe Sweep voltage.
PRESS1 - Pasp box1 pressure sensor.
IV1-V,IV2-V,IV3-V - IV #s 1,2,3 Voltage channels.
CAL2A, CAL2B - A/D #2 MUX A,B Calibration channels.
TEMP2-2 - Pasp box2 temperature sensor #2.
LP-TEMP - Langmuir Probe Temperature.
LP-NE - Langmuir Probe electron density. (Ne)
TEMP1-1 - Pasp box1 temperature sensor #1.
IV1-I,IV2-I,IV3-I - IV #s 1,2,3 Current channels.
RTD-CAL1,RTD-CAL2 - RTD Calibration.

```

14

0 \ Asis Interface definitions - Arrays and control bits15FEB90CDB

```

1 HEX
2
3 1  CONSTANT SP      2  CONSTANT GP      3  CONSTANT CC
4 4  CONSTANT SC      5  CONSTANT TBP     6  CONSTANT TBC
5
6 00 CONSTANT 00-SEL    10 CONSTANT IV-SEL
7 08 CONSTANT FRAME-GND 18 CONSTANT FRAME-NEG
8 00 CONSTANT +V        20 CONSTANT -V
9 00 CONSTANT EMIT-OFF  40 CONSTANT EMIT-ON
10
11 DECIMAL
12
13
14
15

```

164

15FEB90CDB \ Asis Interface definitions - Arrays and control

```

SP - Silicon Planar Array.
GP - GaAs Planar Array.
CC - Cassigranian Concentrator Array.
SC - Silicon Concentrator Array.
TBP - TBD Planar Array.
TBC - TBD Concentrator Array.
00-SEL - ASIS Function disabled.
IV-SEL - ASIS Function = IV.
FRAME-GND - ASIS Function HV array frame grounded.
FRAME-NEG - ASIS Function HV array frame shorted to neg.
+V - Select +HV.
-V - Select -HV.
EMIT-OFF - Disable emitter heater current.
EMIT-ON - Enable emitter heater current.

```

Forth 83 Model

15

0 \ High Voltage Sequence Programming - Orbits 1 - 2
15FEB90CDB

```

1
2
3 SP  FRAME-GND  EMIT-OFF  + +  CONSTANT  OR1-EL1-CMD
4 CC  FRAME-GND  EMIT-OFF  + +  CONSTANT  OR1-EL2-CMD
5 SC  FRAME-GND  EMIT-OFF  + +  CONSTANT  OR1-EL3-CMD
6 TBC FRAME-GND  EMIT-OFF  + +  CONSTANT  OR1-EL4-CMD
7
8
9 SP  FRAME-GND  EMIT-ON   + +  CONSTANT  OR2-EL1-CMD
10 CC FRAME-GND  EMIT-ON   + +  CONSTANT  OR2-EL2-CMD
11 SC FRAME-GND  EMIT-ON   + +  CONSTANT  OR2-EL3-CMD
12 TBC FRAME-GND  EMIT-ON   + +  CONSTANT  OR2-EL4-CMD
13
14
15

```

165

17AUG89CDB \ High Voltage Sequence Programming - Orbits 1 - 2

```

OR1-EL1-CMD - Orbit 1 element 1 command.
OR1-EL2-CMD - Orbit 1 element 2 command.
OR1-EL3-CMD - Orbit 1 element 3 command.
OR1-EL4-CMD - Orbit 1 element 4 command.

OR2-EL1-CMD - Orbit 2 element 1 command.
OR2-EL2-CMD - Orbit 2 element 2 command.
OR2-EL3-CMD - Orbit 2 element 3 command.
OR2-EL4-CMD - Orbit 2 element 4 command.

```

16

0 \ High Voltage Sequence Programming - Orbits 3 - 4
15FEB90CDB

```

1
2
3 CC  FRAME-NEG  EMIT-ON   + +  CONSTANT  OR3-EL1-CMD
4 SC  FRAME-NEG  EMIT-ON   + +  CONSTANT  OR3-EL2-CMD
5 TBP FRAME-GND  EMIT-ON   + +  CONSTANT  OR3-EL3-CMD
6 TBC FRAME-NEG  EMIT-ON   + +  CONSTANT  OR3-EL4-CMD
7
8
9 SP  FRAME-GND  EMIT-ON   + +  CONSTANT  OR4-EL1-CMD
10 SP FRAME-GND  EMIT-ON   + +  CONSTANT  OR4-EL2-CMD
11 GP  FRAME-GND  EMIT-ON   + +  CONSTANT  OR4-EL3-CMD
12 TBP FRAME-GND  EMIT-ON   + +  CONSTANT  OR4-EL4-CMD
13
14
15

```

166

17AUG89CDB \ High Voltage Sequence Programming - Orbits 3 - 4

```

OR3-EL1-CMD - Orbit 3 element 1 command.
OR3-EL2-CMD - Orbit 3 element 2 command.
OR3-EL3-CMD - Orbit 3 element 3 command.
OR3-EL4-CMD - Orbit 3 element 4 command.

OR4-EL1-CMD - Orbit 4 element 1 command.
OR4-EL2-CMD - Orbit 4 element 2 command.
OR4-EL3-CMD - Orbit 4 element 3 command.
OR4-EL4-CMD - Orbit 4 element 4 command.

```

17

0 \ High Voltage Sequence Programming - Orbits 5 - 6
15FEB90CDB

```

1
2
3 CC  FRAME-NEG  EMIT-OFF  + +  CONSTANT  OR5-EL1-CMD
4 SC  FRAME-NEG  EMIT-OFF  + +  CONSTANT  OR5-EL2-CMD
5 TBP FRAME-GND  EMIT-OFF  + +  CONSTANT  OR5-EL3-CMD
6 TBC FRAME-NEG  EMIT-OFF  + +  CONSTANT  OR5-EL4-CMD
7
8
9 GP  FRAME-GND  EMIT-OFF  + +  CONSTANT  OR6-EL1-CMD
10 CC FRAME-GND  EMIT-OFF  + +  CONSTANT  OR6-EL2-CMD
11 SC FRAME-GND  EMIT-OFF  + +  CONSTANT  OR6-EL3-CMD
12 TBC FRAME-GND  EMIT-OFF  + +  CONSTANT  OR6-EL4-CMD
13
14 SP  FRAME-GND  EMIT-OFF  + +  CONSTANT  OCC-CMD
15

```

167

17AUG89CDB \ High Voltage Sequence Programming - Orbits 5 - 6

```

OR5-EL1-CMD - Orbit 5 element 1 command.
OR5-EL2-CMD - Orbit 5 element 2 command.
OR5-EL3-CMD - Orbit 5 element 3 command.
OR5-EL4-CMD - Orbit 5 element 4 command.

OR6-EL1-CMD - Orbit 6 element 1 command.
OR6-EL2-CMD - Orbit 6 element 2 command.
OR6-EL3-CMD - Orbit 6 element 3 command.
OR6-EL4-CMD - Orbit 6 element 4 command.

OCC-CMD - Occultation command.

```

Forth 83 Model

18

0 \ High Voltage Sequence Programming - TEST SEG
15FEB90CDB

```

1
2 SP FRAME-GND  EMIT-ON  + +  CONSTANT  TEST1-CMD
3 GP FRAME-GND  EMIT-ON  + +  CONSTANT  TEST2-CMD
4 CC FRAME-GND  EMIT-ON  + +  CONSTANT  TEST3-CMD
5 CC FRAME-NEG  EMIT-ON  + +  CONSTANT  TEST4-CMD
6 SC FRAME-GND  EMIT-ON  + +  CONSTANT  TEST5-CMD
7 SC FRAME-NEG  EMIT-ON  + +  CONSTANT  TEST6-CMD
8 TBP FRAME-GND  EMIT-ON  + +  CONSTANT  TEST7-CMD
9 TBC FRAME-GND  EMIT-ON  + +  CONSTANT  TEST8-CMD
10 TBC FRAME-NEG  EMIT-ON  + +  CONSTANT  TEST9-CMD

```

```

11
12
13
14
15

```

19

0 \ DACS High Voltage Commands
15FEB90CDB

```

1 DECIMAL
2 31 CONSTANT HV-CAL
3 HV-CAL 000 500 */ +V + CONSTANT +000V
4 HV-CAL 50 500 */ +V + CONSTANT +50V
5 HV-CAL 100 500 */ +V + CONSTANT +100V
6 HV-CAL 150 500 */ +V + CONSTANT +150V
7 HV-CAL 300 500 */ +V + CONSTANT +300V
8 HV-CAL 500 500 */ +V + CONSTANT +500V
9 HV-CAL 000 500 */ -V + CONSTANT -000V
10 HV-CAL 150 500 */ -V + CONSTANT -150V
11 HV-CAL 200 500 */ -V + CONSTANT -200V
12 HV-CAL 300 500 */ -V + CONSTANT -300V
13 HV-CAL 500 500 */ -V + CONSTANT -500V

```

```

14
15

```

20

0 \ DACS High Voltage Commands
15FEB90CDB

```

1 HEX
2 : (HV-TBL) (S addr -- )
3 +000V , +50V , +100V , +150V , +300V , +500V , -000V ,
addr.
4 -150V , -200V , -300V , -500V , ;
5
6 : (HV-C-TBL) (S addr -- )
addr.
7 OR1-EL1-CMD , OR1-EL2-CMD , OR1-EL3-CMD , OR1-EL4-CMD ,
8 OR2-EL1-CMD , OR2-EL2-CMD , OR2-EL3-CMD , OR2-EL4-CMD ,
9 OR3-EL1-CMD , OR3-EL2-CMD , OR3-EL3-CMD , OR3-EL4-CMD ,
10 OR4-EL1-CMD , OR4-EL2-CMD , OR4-EL3-CMD , OR4-EL4-CMD ,
11 OR5-EL1-CMD , OR5-EL2-CMD , OR5-EL3-CMD , OR5-EL4-CMD ,
12 OR6-EL1-CMD , OR6-EL2-CMD , OR6-EL3-CMD , OR6-EL4-CMD ,
13 TEST1-CMD , TEST2-CMD , TEST3-CMD , TEST4-CMD , TEST5-CMD ,
14 TEST6-CMD , TEST7-CMD , TEST8-CMD , TEST9-CMD , ;
15 DECIMAL

```

168

17AUG89CDB \ High Voltage Sequence Programming - TEST SEG

```

TEST1-CMD - Step 1 of Test sequences.
TEST2-CMD - Step 2 of Test sequences.
TEST3-CMD - Step 3 of Test sequences.
TEST4-CMD - Step 4 of Test sequences.
TEST5-CMD - Step 5 of Test sequences.
TEST6-CMD - Step 6 of Test sequences.
TEST7-CMD - Step 7 of Test sequences.
TEST8-CMD - Step 8 of Test sequences.
TEST9-CMD - Step 9 of Test sequences.

```

169

14SEP89CDB \ DACS High Voltage Commands

```

DECIMAL
HV-CAL - ASIS code required for 500 volts out.
+000V - ASIS command for +000 volts.
+50V - ASIS command for +50 volts.
+100V - ASIS command for +100 volts.
+150V - ASIS command for +150 volts.
+300V - ASIS command for +300 volts.
+500V - ASIS command for +500 volts.
-000V - ASIS command for -000 volts.
-150V - ASIS command for -150 volts.
-200V - ASIS command for -200 volts.
-300V - ASIS command for -300 volts.
-500V - ASIS command for -500 volts.

```

170

14SEP89CDB \ DACS High Voltage Commands

```

HEX
: (HV-TBL) (S addr -- )
When executed it builds a ASIS voltage command table at

: (HV-C-TBL) (S addr -- )
When executed it builds a ASIS control command table at

```


21

0 \ IV SUPPORT - IV TABLES 1 - 3

15FEB90CDB

1 HEX

2 : IV1-RES (S --)

3 000 000 000 000 000 000 080 040 0C0 020 0A0 060 0E0 010 090 050

addr.

4 0D0 030 0B0 070 008 088 0C8 0A8 018 038 084 0E4 04C 042 026 0E3

5 SP IV ;

6 : IV2-RES (S --)

7 000 000 000 000 000 0C0 020 0A0 060 0E0 010 090 050 0D0 030 0B0

addr.

8 070 0F0 008 088 0C8 0A8 0E8 058 078 044 014 00C 0BC 08A 05E 02F

9 GP IV ;

10 : IV3-RES (S --)

11 000 000 080 040 0C0 020 0A0 060 0E0 010 090 050 0D0 030 0B0 070

addr.

12 0F0 008 048 0C8 0A8 0E8 058 0B8 004 0A4 034 0AC 042 09A 0A1 0FF

13 CC IV ;

14 DECIMAL

15

171

16MAR90CDB \ IV SUPPORT - IV TABLES 1 - 3

HEX

: IV1-RES (S --)

When executed it builds a IV1 resistor control table at

: IV2-RES (S --)

When executed it builds a IV2 resistor control table at

: IV3-RES (S --)

When executed it builds a IV3 resistor control table at

22

0 \ IV SUPPORT - IV TABLES 4 - 6

15FEB90CDB

1 HEX

2 : IV4-RES (S --)

3 000 000 080 040 0C0 020 0A0 060 0E0 010 090 050 0D0 030 0B0 070

addr.

4 0F0 088 048 028 068 018 058 078 084 0E4 0B4 06C 022 0DA 091 0FF

5 SC IV ;

6 : IV5-RES (S --)

7 000 000 000 000 000 000 080 040 0C0 020 0A0 060 0E0 010 090 050

addr.

8 0D0 030 0B0 070 008 088 0C8 0A8 018 038 084 0E4 04C 042 026 0E3

9 TBP IV ;

10 : IV6-RES (S --)

11 000 000 080 040 0C0 020 0A0 060 0E0 010 090 050 0D0 030 0B0 070

addr.

12 0F0 008 048 0C8 0A8 0E8 058 0B8 004 0A4 034 0AC 042 09A 0A1 0FF

13 TBC IV ;

14 DECIMAL

15

172

16MAR90CDB \ IV SUPPORT - IV TABLES 4 - 6

HEX

: IV4-RES (S --)

When executed it builds a IV4 resistor control table at

: IV5-RES (S --)

When executed it builds a IV5 resistor control table at

: IV6-RES (S --)

When executed it builds a IV6 resistor control table at

23

0 \ Pasp test plug definitions

15FEB90CDB

1 HEX

2

3 40 CONSTANT TST-BIT

4 20 CONSTANT TSS-BIT

5 60 CONSTANT DEV-BIT

6

7 DECIMAL

8

9

10

11

12

13

14

15

173

6JUL90CDB \ Pasp test plug definitions

HEX

TST-BIT - Hardware interlock for test sequence.

TSS-BIT - Hardware interlock for short test.

DECIMAL

Forth 83 Model

24

0 \ Langmuir Probe Support
 15FEB90CDB
 1 HEX
 2
 3 80 CONSTANT LP-ON-CMD
 4 C8 CONSTANT LP-RATE
 5 LP-RATE 2* CONSTANT LP-SAMPLES
 6
 7 DECIMAL
 8
 9
 10
 11
 12
 13
 14
 15

25

0 \ DACS CHANNEL ASSIGNMENTS - POWER MONITORS
 15FEB90CDB
 1 HEX
 2
 3 01 CONSTANT 5V-BOX1 02 CONSTANT +15V-BOX1
 4 04 CONSTANT -15V-BOX1 08 CONSTANT 28V-RELAY
 5 10 CONSTANT 28V-HVPS 20 CONSTANT 5V-EMIT
 6 40 CONSTANT 15AD210 80 CONSTANT PM1-7
 7
 8 01 CONSTANT 5VAK 02 CONSTANT 5VMO
 9 04 CONSTANT 5VLO 08 CONSTANT 5VDAC
 10 10 CONSTANT +15VDAC 20 CONSTANT -15VDAC
 11 40 CONSTANT 28V1 80 CONSTANT PM2-7
 12
 13
 14 DECIMAL
 15

26

0 \ PASP COMMAND DEFINITION
 15FEB90CDB
 1 HEX
 2 00 CONSTANT SPR1-CMD 01 CONSTANT ORB1-CMD
 3 02 CONSTANT ORB2-CMD 03 CONSTANT ORB3-CMD
 4 04 CONSTANT ORB4-CMD 05 CONSTANT ORB5-CMD
 5 06 CONSTANT ORB6-CMD 07 CONSTANT GARR-CMD
 6 08 CONSTANT STBY-CMD 09 CONSTANT EXEC-CMD
 7 0A CONSTANT CSEL-CMD 0B CONSTANT PSEL-CMD
 8 0C CONSTANT TPMM-CMD 0D CONSTANT LPWR-CMD
 9 0E CONSTANT MRES-CMD 0F CONSTANT SPR2-CMD
 10
 11 0 CONSTANT CMD-FLG -1 CONSTANT TPM-FLG
 12 -2 CONSTANT PSEL-FLG -3 CONSTANT CSEL-FLG
 13 30 CONSTANT CMD-LIMIT
 14 DECIMAL
 15

174

20MAR90CDB \ Langmuir Probe Support

HEX

LP-ON-CMD - DIG bit to control the Langmuir probe command.
 LP-RATE - Langmuir Probe sample rate (Samples/sec.)
 LP-SAMPLES - Number of Langmuir Probe samples (2 secs.)

DECIMAL

175

24AUG89CDB \ DACS CHANNEL ASSIGNMENTS - POWER MONITORS

HEX

Not used

DECIMAL

176

20MAR90CDB \ PASP COMMAND DEFINITION

SPR1-CMD - Spare command #1.
 ORB1-CMD, ORB2-CMD, ORB3-CMD, ORB4-CMD, ORB5-CMD, ORB6-CMD - Start orbits 1-6 commands.
 GARR-CMD - Start Garrett mode command.
 STBY-CMD - Start Standby mode command.
 EXEC-CMD - Execute routine pointed to by CSEL parameter.
 CSEL-CMD - Set CSEL parameter command (2 byte command.)
 PSEL-CMD - Set PSEL parameter command (2 byte command.)
 TPMM-CMD - Send TPM COMMAND command (2 byte command.)
 LPWR-CMD - Dacs power off (Low power mode) command.
 MRES-CMD - Dacs power on or manual reset command.
 SPR2-CMD - Spare command #2.
 CMD-FLG, TPM-FLG, PSEL-FLG, CSEL-FLG, CMD-LIMIT - Command handler flags.

Forth 83 Model

27

0 \ A/D Time constants

15FEB90CDB

1 HEX

2

3 0A CONSTANT AD-TIME

4

5 25 CONSTANT AD-DELAY

6

7 21 CONSTANT CAL-TC

8 DECIMAL

9

10

11

12

13

14

15

28

0 \ E-BUFFER SUPPORT ERROR CODES

16FEB90CDB

1 HEX

2 01 CONSTANT TEL-ERRO

;

3 02 CONSTANT MET-ERRO

4 03 CONSTANT CMD-ERRO

5 04 CONSTANT CMD-ERR1

6 05 CONSTANT AD1-ERRO

7 06 CONSTANT AD2-ERRO

8 07 CONSTANT RTD-ERRO

9 08 CONSTANT SUM-ERR

;

10 09 CONSTANT NOOP-ERR

;

11 0A CONSTANT COLD-ERR

;

12 0B CONSTANT R-MAX-ERR

DROP ;

13 DECIMAL

14

15

29

0 \ System counters

16FEB90CDB

1 VARIABLE RES-CNT VARIABLE WDG-CNT VARIABLE ORB-CNT

2 VARIABLE SEG-CNT VARIABLE ELE-CNT

RES-CNT,WDG-CNT,ORB-CNT,SEG-CNT,ELE-CNT,SBY-CNT,GAR-CNT,OCC-CNT

3 VARIABLE GAR-CNT VARIABLE OCC-CNT VARIABLE TSS-CNT

4 VARIABLE TST-CNT VARIABLE TIM-CNT

5

6 VARIABLE HSK-CNT VARIABLE IVS-CNT VARIABLE HVS-CNT
buffer.

7 VARIABLE TPH-CNT VARIABLE LMP-CNT VARIABLE CAL-CNT

8

9 VARIABLE CMD-CNT VARIABLE INT-CNT VARIABLE TPM-CNT

10 VARIABLE MET-CNT VARIABLE OCI-CNT

11 VARIABLE NEW-PAR VARIABLE NEW-CAL VARIABLE NEW-CMD

12

13 VARIABLE RP-FLG VARIABLE M-TEMP VARIABLE E-TEMP

14 VARIABLE RP-TEMP

15

177

15FEB90CDB \ A/D Time constants

HEX

AD-TIME - Time required for a/d conversion.

AD-DELAY - Time required for switched a/d channel to settle.

CAL-TC - Time between samples for calibration data words.

DECIMAL

178

16FEB90CDB \ E-BUFFER SUPPORT ERROR CODES -- GSE WORDS

HEX

: TEL-ERRO (S n --) ." Telemetry pointer over-run " DROP

: MET-ERRO (S n --) ." Met increment overflow " DROP ;

: CMD-ERRO (S n --) ." Command receive error " H. ;

: CMD-ERR1 (S n --) ." Command decode error " H. ;

: AD1-ERRO (S n --) ." AD1 Conversion error " H. ;

: AD2-ERRO (S n --) ." AD2 Conversion error " H. ;

: RTD-ERRO (S n --) ." RTD Conversion error " H. ;

: SUM-ERR (S n --) ." Stack recovery failure - SUM " DROP

: NOOP-ERR (S n --) ." Stack recovery failure - NOOP " DROP

: COLD-ERR (S n --) ." Stack recovery failure - COLD " DROP

: R-MAX-ERR (S n --) ." Stack recovery failure - R-MAX "

DECIMAL

179

18JUL90CDB \ System counters

Event marker counters

VARIABLE SBY-CNT

TSS-CNT,TST-CNT.

Data marker counters

HSK-CNT,IVS-CNT,HVS-CNT,TPH-CNT,LMP-CNT,CAL-CNT.

These are incremented when a marker is added to the DATA

Event counters

CMD-CNT,INT-CNT,TPM-CNT,MET-CNT,OCI-CNT.

Command parameters

NEW-PAR,NEW-CAL,NEW-CMD.

Misc engineering parameters

RP-FLG,M-TEMP,E-TEMP,RP-TEMP.

Forth 83 Model

30

0 \ VARIABLE DEFINITIONS

11FEB90C16FEB90CDB

1 VARIABLE FRAME-CNT

2 VARIABLE RDO-CRC

VARIABLE TDO-CRC

3 VARIABLE CMD-STAT

VARIABLE DATA-STAT

4

5 VARIABLE MCLK 4 ALLOT-R

6 VARIABLE MET-TEMP 4 ALLOT-R

bits.

7 VARIABLE MET-SET 4 ALLOT-R

8 VARIABLE MET-SYNC

9

10 VARIABLE PASS-CNT

11 2VARIABLE PCLK

12 VARIABLE STEP

Telemetry.

13 VARIABLE 4.1-CNTR

14 VARIABLE TDO-FLG

ports

15 VARIABLE MET-IMAGE VARIABLE RDO-IMAGE VARIABLE DIG-IMAGE

31

0 \ Buffer configuration

16FEB90CDB

1 HEX

2 VARIABLE BUF-START BUF-SIZE ALLOT-R

3 2VARIABLE BUF-WRT 2VARIABLE BUF-RD

4 VARIABLE BUF-GAUGE 2VARIABLE PTR-RD

5

6 VARIABLE E-BUF-START E-BUF-SIZE ALLOT-R

7 2VARIABLE E-BUF-WRT 2VARIABLE E-BUF-RD

8 VARIABLE E-BUF-GAUGE

9

10

11

12

13

14

15

32

0 \ Protected Memory Support -

16FEB90CDB

1 HEX

2

3 VARIABLE WDOG-FLG VARIABLE SAVE-R?

4 VARIABLE P-ADDR VARIABLE P-DATA

5 VARIABLE P-FLG VARIABLE P-WRT

6

7 VARIABLE HV-PTR (POINTER TO HV-C-TBL)

8 VARIABLE LP-CNTR

9 VARIABLE LP-FLG

10

11 VARIABLE CMD-PTR

12 VARIABLE CMD-CLK

13 VARIABLE RDO-STAMP VARIABLE CMD-STAMP

14 DECIMAL

15

180

16FEB90CDB \ DEFINITIONS

FRAME-CNT - Telemetry frame count.

RDO-CRC,TDO-CRC - Temp storage for CRC calculations.

CMD-STAT,DATA-STAT - Not used.

MCLK,MET-TEMP,MET-SET - Mission elapsed time registers 48

MET-SYNC - MET sync flag. Move MET-SET to MCLK.

PASS-CNT - NMI counter 2.5ms/count.

PCLK - Pasp system clock 5ms. counter 32 bits.

STEP - Telemetry scheduler.

4.1-CNTR - 4.1 second software backup to RDO driven

TDO-FLG - Used to schedule TDO dump request.

MET-IMAG,RDO-IMAG,DIG-IMAGE - Temp storage for write only

181

16FEB90CDB \ Buffer configuration

HEX

BUF-START - Start of data buffer.

BUF-WRT - Pointer used to write to data buffer.

BUF-RD - Pointer used to read from data buffer.

BUF-GAUGE - Incremented on write, decremented on read.

PTR-RD - Pointer is passed in the telemetry.

E-BUF-START - Start of error buffer.

E-BUF-WRT - Pointer used to write to error buffer.

E-BUF-RD - Pointer used to read from error buffer.

E-BUF-GAUGE - Incremented on write, decremented on read.

Pointers are double words. Lower word is buffer address and upper words is buffer page count.

182

18JUL90CDB \ Protected Memory Support -

WDOG-FLG - N.U.

SAVE-R? - N.U.

P-ADDR - N.U.

P-DATA - N.U.

P-FLG - N.U.

P-WRT - N.U.

HV-PTR - Pointer to HV Command table.

LP-CNTR - Counter to schedule LP data sequence.

LP-FLG - Flag to start and end LP data sequence.

CMD-PTR - Pointer to control command decoder mode.

CMD-CLK - Default timer to end multi-word commands.

Forth 83 Model

33

```

0 \ Pointers to rom tables - Start of Pasp code
16FEB90CDB
1
2 0 CONSTANT TEL-TBL ( Telemetry schedule )
3 0 CONSTANT CMD-TBL ( Command definition )
4 0 CONSTANT CAL-TBL ( Cal routine definition )
5 0 CONSTANT HV-TBL ( HV command definition )
6 0 CONSTANT HV-C-TBL ( Orbit/element definition )
7
8
9
10
11
12
13
14
15

```

34

```

0 \ Dacs telemetry support - Telemetry crc calculation
2FEB88CDB
1 HEX
2
3 LABEL 1-BIT
4 DX BX MOV
5 DX SHL AH AL AND 00 # AL CMP 0<> IF DX INC THEN
6 8000 # BX TEST 0<>
7 IF 1021 # DX XOR THEN RET
8
9 LABEL CRC-CALC
10 AX PUSH BX PUSH CX DX MOV 80 # AH MOV
11 8 DO AX PUSH 1-BIT #) CALL AX POP AH SHR LOOP
12 BX POP AX POP RET
13
14 DECIMAL
15

```

35

```

0 \ Dacs telemetry support - Real time telemetry data
18JAN88CDB
1 HEX
2 LABEL RDO-RAM1
3 AL RDO-RAM [DI] BYTE MOV AL RDO-RAM [DI] BYTE CMP
4 0<> IF AL RDO-RAM [DI] BYTE MOV THEN RET
5
6 LABEL R-SEND
7 RDO-RAM1 #) CALL
8 RDO-CRC #) CX MOV CRC-CALC #) CALL DX RDO-CRC #) MOV
9 AH AL MOV
10
11 DI INC RDO-RAM1 #) CALL DI DEC
12 RDO-CRC #) CX MOV CRC-CALC #) CALL DX RDO-CRC #) MOV
13 RET
14 DECIMAL
15

```

183

```

16FEB90CDB \ Pointers to rom tables - Start of Pasp code
TEL-TBL - Pointer to table. ( Telemetry schedule )
CMD-TBL - Pointer to table. ( Command definition )
CAL-TBL - Pointer to table. ( Cal routine definition )
HV-TBL - Pointer to table. ( HV command definition )
HV-C-TBL - Pointer to table. ( Orbit/element definition )

```

184

```

16SEP89CDB \ Dacs buffer support - crc calculation
HEX
1-BIT is called from CRC-CALC .
in - AH = bit mask byte
in - DX = current crc calculation - lost
crc calculation performed with X16 + X12 + X5 + 1 polynomial
out - DX = new crc calculation

CRC-CALC is used to calculate a new crc code .
in - CX = current crc calculation - lost
in - AL = output byte
in - AH is used - lost
in - BX is used
out - DX = new crc calculation

DECIMAL

```

185

```

2FEB88CDB \ REAL TIME TELEMETRY SUPPORT
RDO-RAM1 - write to real time buffer, check for error .
in - AL = output byte
in - DI = pointer to real time buffer

R-SEND - send a byte to the real time controller buffer
in - AL = output byte - lost
in - DI = pointer to real time buffer
in - CX is used - lost
in - DX is used - lost

```

Forth 83 Model

```

36
0 \ Dacs telemetry support - Tape telemetry data
18JAN88CDB
1 HEX
2 LABEL TDO-RAM!
3 AL TDO-RAM [DI] BYTE MOV AL TDO-RAM [DI] BYTE CMP
4 0< IF AL TDO-RAM [DI] BYTE MOV THEN RET
5
6 LABEL T-SEND
7 TDO-RAM! #) CALL
8 TDO-CRC #) CX MOV CRC-CALC #) CALL DX TDO-CRC #) MOV
9 AH AL MOV
10
11 DI INC TDO-RAM! #) CALL DI DEC
12 TDO-CRC #) CX MOV CRC-CALC #) CALL DX TDO-CRC #) MOV
13 RET
14 DECIMAL
15

```

```

37
0 \ Dacs telemetry support - Telemetry crc output
18JAN88CDB
1 HEX
2
3 LABEL R-CRC
4 DO # AL MOV RDO-CRC #) CX MOV CRC-CALC #) CALL
5 DX CX MOV CRC-CALC #) CALL DX AX MOV
6 RDO-RAM! #) CALL AH AL MOV
7 DI INC RDO-RAM! #) CALL DI DEC RET
8
9 LABEL T-CRC
10 DO # AL MOV TDO-CRC #) CX MOV CRC-CALC #) CALL
11 DX CX MOV CRC-CALC #) CALL DX AX MOV
12 TDO-RAM! #) CALL AH AL MOV
13 DI INC TDO-RAM! #) CALL DI DEC RET
14 DECIMAL
15

```

```

38
0 \ Dacs telemetry support - Telemetry data output
18JAN88CDB
1 HEX
2
3 LABEL SEND-AX
4 AX PUSH R-SEND #) CALL
5 AX POP T-SEND #) CALL RET
6
7 DECIMAL
8
9
10
11
12
13
14
15

```

```

186
2FE88CDB \ TAPE RECORDER TELEMETRY SUPPORT

```

TDO-RAM! - write to tape buffer, check for error .
in - AL = output byte
in - DI = pointer to tape buffer

T-SEND - send a byte to the tape controller buffer .
in - AL = output byte - lost
in - DI = pointer to tape buffer
in - CX is used - lost
in - DX is used - lost

```

187
4FE88CDB \ REAL TIME TELEMETRY SUPPORT

```

R-CRC - flush crc register and save RDO crc calculation
in - AX is used - lost
in - CX is used - lost
in - DX is used - lost
in - DI = pointer to real time buffer

T-CRC - flush crc register and save TDO crc calculation
in - AX is used - lost
in - CX is used - lost
in - DX is used - lost
in - DI = pointer to tape buffer

```

188
2FE88CDB \ TAPE RECORDER TELEMETRY SUPPORT

```

SEND-AX - send AX to both telemetry buffers .
in - AX is used - lost
in - CX is used - lost
in - DX is used - lost
in - DI = pointer to real time buffer

Forth 83 Model

39

0 \ Dacs buffer support - Buffer initialization
16SEP89CDB

1 HEX

2

3 LABEL INIT-BUF

buffer

4 00 # BUF-WRT #) MOV 00 # BUF-WRT 2 + #) MOV

5 00 # BUF-RD #) MOV 00 # BUF-RD 2 + #) MOV

6 00 # BUF-GAUGE #) MOV

7 BUF-SIZE DO CX DI MOV DI DEC DI DEC

8 B-MT # BUF-START [DI] MOV CX DEC LOOP

9 RET

10

11 DECIMAL

12

13

14

15

40

0 \ Dacs buffer support - Buffer input words
16SEP89CDB

1 HEX

2 LABEL WRT-BUFR

3 BUF-WRT #) DI MOV AL BUF-START [DI] MOV

4 DI INC BUF-SIZE # DI CMP

5 U>= IF 00 # DI MOV BUF-WRT 2+ #) INC THEN

6 DI BUF-WRT #) MOV BUF-GAUGE #) INC

7 RET

8

9 DECIMAL

10

11

12

13

14

15

41

0 \ Dacs buffer support - Buffer output words
18JAN88CDB

1 HEX

2

3 LABEL RD-BUF

4 PTR-RD #) DI MOV BUF-START [DI] AX MOV

5 DI INC DI INC BUF-SIZE # DI CMP

6 U>= IF 00 # DI MOV PTR-RD 2+ #) INC THEN

7 DI PTR-RD #) MOV

8 RET

9

10 DECIMAL

11

12

13

14

15

189

16SEP89CDB \ BUFFER SUPPORT

INIT-BUF - Read and write pointers set to top of page 0

Set byte count to 0. Initialize the dacs data buffer to
ascii characters MT.

in - CX is used - lost

in - DI is used - lost

190

16SEP89CDB \ BUFFER SUPPORT

WRT-BUFR - Write a byte into the DACS buffer at location
pointed to by BUF-WRT offset.Increment the BUF-WRT modulo BUF-SIZE and increment
buffer page (BUF-WRT + 2) as required.

Increment BUF-GAUGE.

in - AL is used

in - DI is used - lost

191

2FEB88CDB \ BUFFER SUPPORT

RD-BUF - Read a word from the DACS buffer at location
pointed to by PTR-RD offset.Point to next word modulo BUF-SIZE and increment
buffer page (PTR-RD + 2) as required.

in - DI is used - lost

out - AX = word out

Forth 83 Model

42

0 \ Dacs buffer support - Buffer output initialization 2FEB88CDB \ BUFFER SUPPORT
18JAN88CDB

```
1 HEX
2 LABEL INIT-RD
3 DATA-BYTES # BUF-GAUGE #) CMP U>=
4 IF BUF-RD #) AX MOV AX PTR-RD #) MOV
5 BUF-RD 2 + #) AX MOV AX PTR-RD 2 + #) MOV
6 DATA-BYTES # BUF-GAUGE #) SUB
7 DATA-BYTES # BUF-RD #) ADD BUF-SIZE # BUF-RD #) CMP U>=
8 IF BUF-SIZE # BUF-RD #) SUB BUF-RD 2 + #) INC THEN
9 ELSE BUF-WRT #) AX MOV AX PTR-RD #) MOV
10 BUF-WRT 2 + #) AX MOV AX PTR-RD 2 + #) MOV
11 DATA-BYTES # PTR-RD #) SUB 00 # PTR-RD #) CMP <
12 IF BUF-SIZE # PTR RD #) ADD PTR-RD 2 + #) DEC THEN
13 8000 # PTR-RD #) OR
14 THEN RET DECIMAL
15
```

192

INIT-RD - IF dacs buffer contains enough data to fill a frame
send a full frame, adjust BUF-GAUGE and move BUF-RD .
ELSE send latest data and previous data as filler . Mark as
partial frame.
in - AX is used - lost

43

0 \ Error buffer support - Error buffer initialization 16SEP89CDB \ E-BUFFER SUPPORT
16FEB90CDB

```
1 HEX
2
3 LABEL INIT-E-BUF
4 00 # E-BUF-WRT #) MOV 00 # E-BUF-WRT 2 + #) MOV
5 00 # E-BUF-RD #) MOV 00 # E-BUF-RD 2 + #) MOV
to
6 00 # E-BUF-GAUGE #) MOV
7 E-BUF-SIZE 00 CX DI MOV DI DEC DI DEC
8 B-MT # E-BUF-START [DI] MOV CX DEC LOOP
9 RET
10
11 DECIMAL
12
13
14
15
```

193

INIT-E-BUF - Read and write pointers set to top of
page 0 buffer 0 .
Set byte count to 0 . Initialize the dacs error buffer
ascii characters MT .
in - CX is used - lost
in - DI is used - lost

44

0 \ Error buffer support - Error buffer input words 16SEP89CDB \ E-BUFFER SUPPORT
16SEP89CDB

```
1 HEX
2
3 LABEL SYS-ERR
4 E-BUF-WRT #) DI MOV AX E-BUF-START [DI] MOV
5 DI INC DI INC E-BUF-SIZE # DI CMP
6 >= IF 00 # DI MOV E-BUF-WRT 2+ #) INC THEN
7 DI E-BUF-WRT #) MOV E-BUF-GAUGE #) INC
8 RET
9
10
11 DECIMAL
12
13
14
15
```

194

SYS-ERR - Write a word into the ERROR buffer at location
pointed to by E-BUF-WRT offset .
Increment the E-BUF-WRT modulo E-BUF-SIZE and increment
buffer page (E-BUF-WRT + 2) as required .
Increment BUF-GAUGE .
in - AL = error data
in - AH = error code
in - DI is used - lost

Forth 83 Model


```

45
0 \ Error buffer support - Error buffer output words
16SEP89CDB
1 HEX
2
3 LABEL RD-E-BUF
4 00 # E-BUF-GAUGE #) CMP >
5 IF E-BUF-GAUGE #) DEC E-BUF-RD #) DI MOV
6 E-BUF-START [DI] AX MOV
7 DI INC DI INC E-BUF-SIZE # DI CMP
8 >= IF 00 # DI MOV E-BUF-RD 2+ #) INC THEN
9 DI E-BUF-RD #) MOV
10 ELSE FALSE # AX MOV THEN RET
11
12 DECIMAL
13
14
15

```

```

46
0 \ Telemetry data packet support - Exit routine
2FEB88CDB
1 HEX
2 LABEL GET-DI
RDO
3 STEP #) DI MOV 7FFF # DI AND DI DI ADD RET
4
5 LABEL TEL-XIT
6 STEP #) INC RET
7
8 LABEL TEL-XIT1
9 GET-DI #) CALL
10 SEND-AX #) CALL TEL-XIT #) JMP
11
12 LABEL TEL-XIT2
13 AX CX MOV CL AH MOV CH AL MOV TEL-XIT1 #) JMP
14 DECIMAL
15

```

```

47
0 \ Telemetry packet support - Init crc, send ID
2FEB88CDB
1 HEX
2
3 LABEL TEL-00
4 00 # RDO-CRC #) MOV 00 # TDO-CRC #) MOV
5 PACKET-ID # AX MOV TEL-XIT2 #) JMP
6
7 DECIMAL
8
9
10
11
12
13
14
15

```

```

195
16SEP89CDB \ E-BUFFER SUPPORT

```

```

HEX
RD-E-BUF - IF error buffer is not empty decrement E-BUF-GAUGE
and read a word from the ERROR buffer at location
pointed to by E-BUF-RD offset .
Point to next word modulo E-BUF-SIZE and increment
buffer page (E-BUF-RD + 2) as required .
ELSE get the false flag
in - DI is used - lost
out - AX = word out

```

```

196
2FEB88CDB \ NMI interrupt support

```

```

GET-DI - Retrieve STEP, clear the flag bit and double the
count . DI contains the pointer used to store words in
and TDO buffers .
out - DI = buffer pointer .
TEL-XIT - Increment STEP .
TEL-XIT1 - Retrieve pointer .
Send word to RDO and TDO buffers go to TEL-XIT .
in - DI = buffer pointer .
in - AX = word to buffer .
TEL-XIT2
Swap AL and AH go to TEL-XIT1 .
in - CX is used - lost
in - AX = data word .
out - AX = modified data word .

```

```

197
2FEB88CDB \ NMI interrupt support

```

```

TEL-00 - Executed when STEP = 0 .
Initialize crc registers for RDO and TDO data frames .
Send PACKET-ID to both buffers .
in - AX is used - lost .

```

Forth 83 Model

```

48
0 \ Telemetry packet support - Send cnt,len and dest 2FEB88CDB \ NMI interrupt support
2FEB88CDB
1 HEX
2
3 LABEL TEL-01
4 PACKET-CNT # AX MOV FRAME-CNT #) CX MOV 03FFF # CX AND
5 CX AX ADD CX INC CX FRAME-CNT #) MOV TEL-XIT2 #) JMP
6
7 LABEL TEL-02
8 P-LENGTH # AX MOV TEL-XIT2 #) JMP
9
10 LABEL TEL-03
11 GET-DI #) CALL
12 DEST-RDO # AX MOV R-SEND #) CALL
13 DEST-TDO # AX MOV T-SEND #) CALL
14 TEL-XIT #) JMP
15 DECIMAL

```

```

198
TEL-01 - Executed when STEP = 1 .
Make frame count word, increment frame counter and send
frame count word to RDO and TDO buffers .
in - AX is used - lost .
TEL-02 - Executed when STEP = 2 .
Send P-LENGTH to both buffers .
in - AX is used - lost .
TEL-03 - Executed when STEP = 3 .
Retrieve pointer .
Send RDO destination code to RDO buffer .
Send TDO destination code to TDO buffer .
go to TEL-XIT .
in - AX is used - lost .

```

```

49
0 \ Telemetry packet support - Send MET word 1 2FEB88CDB \ NMI interrupt support
2FEB88CDB
1 HEX
2
3 LABEL TEL-04
4 MCLK 4 + #) AX MOV AX MET-TEMP 4 + #) MOV
5 MCLK 2 + #) AX MOV AX MET-TEMP 2 + #) MOV
6 MCLK 0 + #) AX MOV AX MET-TEMP 0 + #) MOV
7 MET-TEMP 4 + #) AX MOV MET-TEMP 2+ #) DX MOV
8 DL SHL DX SHL DX SHL AL RCL DX SHL AL RCL DX SHL AX RCL
9 DX SHL AX RCL DX MET-TEMP 4 + #) MOV TEL-XIT2 #) JMP
10
11 DECIMAL
12
13
14
15

```

```

199
TEL-04 - Executed when STEP = 4 .
Capture current MET count, format first of 3 MET words and
save residual MET data in MET-TEMP + 4 .
Send first MET word to both buffers .
in - AX is used - lost .
in - DX is used - lost .

```

```

50
0 \ Telemetry packet support - Send MET 2, 3 & stat 22FEB88CDB \ NMI interrupt support
2FEB88CDB
1 HEX
2
3 LABEL TEL-05
4 MET-TEMP 4 + #) AX MOV MET-TEMP #) DX MOV
5 4 # CL MOV DX CL SHL 03F # DH AND DH AL OR
6 DX MET-TEMP 4 + #) MOV TEL-XIT2 #) JMP
7
8 LABEL TEL-06
9 MET-TEMP 4 + #) DX MOV DL AH MOV TEL-XIT2 #) JMP
10
11 LABEL TEL-07
12 STATUS-CNT # AX MOV TEL-XIT2 #) JMP
13
14 DECIMAL
15

```

```

200
TEL-05 - Executed when STEP = 5 .
Format the second of 3 MET words and save residual MET data
in MET-TEMP + 4 . Send second MET word to both buffers .
in - AX is used - lost .
in - CX is used - lost .
in - DX is used - lost .
TEL-06 - Executed when STEP = 6 .
Format the third of 3 MET words .
Send third MET word to both buffers .
in - AX is used - lost .
in - DX is used - lost .
TEL-07 - Executed when STEP = 7 .
Send STATUS-CNT to both buffers .
in - AX is used - lost .

```

Forth 83 Model

<p>51 0 \ Telemetry packet support - Send sun sensor data 2FEB88CDB 2FEB88CDB 1 HEX 2 3 LABEL TEL-08 4 SSX #) AL MOV AL SSX #) CMP 0<> data 5 IF SSX #) AL MOV THEN 6 SSY #) AH MOV AH SSY #) CMP 0<> 7 IF SSY #) AH MOV THEN 8 TEL-XIT2 #) JMP 9 data 10 LABEL TEL-09 11 SSA #) AL MOV AL SSA #) CMP 0<> IF SSA #) AL MOV THEN 12 OO # AH MOV TEL-XIT2 #) JMP 13 14 DECIMAL 15</p>	<p>201 \ NMI interrupt support TEL-08 - Executed when STEP = 8 . Collect and format sun sensor x and y data for ancillary word # 1 . Send SS-XY to RDO and TDO buffers . in - AX is used - lost . TEL-09 - Executed when STEP = 9 . Collect and format sun sensor threshold data for ancillary word # 2 . Send SS-ATA to RDO and TDO buffers . in - AX is used - lost .</p>
<p>52 0 \ Telemetry packet support - Send ancillary, pointers 2FEB88CDB 2FEB88CDB 1 HEX 2 3 LABEL TEL-10 4 CMD-STAT #) AX MOV TEL-XIT2 #) JMP 5 6 LABEL TEL-11 7 DATA-STAT #) AX MOV TEL-XIT2 #) JMP 8 9 LABEL TEL-12 bit 10 INIT-RD #) CALL PTR-RD #) AX MOV flag) 11 7FFF # PTR-RD #) AND TEL-XIT1 #) 'MP 12 13 LABEL TEL-13 14 PTR-RD 2 + #) AX MOV TEL-XIT1 #) JMP 15 DECIMAL</p>	<p>202 \ NMI interrupt support TEL-10 - Executed when STEP = 10 . Send CMD-STAT to both buffers ancillary data word # 3 . in - AX is used - lost . TEL-11 - Executed when STEP = 11 . Send DATA-STAT to both buffers ancillary data word # 4 . in - AX is used - lost . TEL-12 - Executed when STEP = 12 . This is the start of data Initialize DACS buffer read pointer and clear partial frame Send dacs buffer pointer with partial frame flag (b15 = in - AX is used - lost . TEL-13 - Executed when STEP = 13 . Send dacs buffer pointer page count . in - AX is used - lost .</p>
<p>53 0 \ Telemetry packet support - Send data and crc words 2FEB88CDB 2FEB88CDB 1 HEX 2 3 LABEL TEL-14 4 RD-BUF #) CALL TEL-XIT1 #) JMP 5 6 LABEL TEL-15 7 GET-DI #) CALL R-CRC #) CALL T-CRC #) CALL 8 -1 # STEP #) MOV 9 TEL-XIT #) JMP 10 11 DECIMAL 12 13 14 15</p>	<p>203 \ NMI interrupt support TEL-14 - Executed when STEP = 14 . Read a word from the DACS buffer and send it to RDO and TDO buffers . This routine is executed 242 times per frame . TEL-15 - Executed when STEP = 15 . Retrieve the telemetry buffer pointer . Send the RDO crc calculation to the RDO buffer . Send the TDO crc calculation to the TDO buffer . Reset STEP to stop frame transfer . Go to TEL-XIT</p>

Forth 83 Model

```

54
0 \ Telemetry packet support - Telemetry packet format 17FEB90CDB \ NMI interrupt support
16SEP89CDB
1 HEX
2 LABEL DACS->TEL
3 OVERHEAD-WORDS 1 - # AL CMP U<
4 IF CS: ' TEL-TBL >BODY #) DI MOV 000F # AX AND
5 AX DI ADD AX DI ADD 0 [DI] JMP
6 ELSE
7 FRAME-WORDS 1 - # AL CMP U<
8 IF TEL-14 #) JMP THEN
9 FRAME-WORDS 1 - # AL CMP 0=
10 IF TEL-15 #) JMP THEN
11 FRAME-WORDS 1 - # AL CMP U>
12 IF TEL-ERRO # AX MOV SYS-ERR #) JMP THEN
13 THEN
14
15 DECIMAL

```

```

204
TEL-N1 - Frame generation algorithm .
IF STEP is pointing to overhead words ( ie. not data )
use the TEL-TBL execution vector to send header etc .
ELSE
IF STEP is pointing to data words send DACS buffer data .
IF STEP is pointing to crc words send crc .
IF STEP is pointing past end of frame post error .
THEN

```

```

55
0 \ NMI interrupt support - Sun timers and Tele cntrl 9JUN89CDB \ NMI interrupt support
16FEB90CDB
1 HEX
2
3 LABEL 4.1-SEC
4 8000 # STEP #) MOV 00 # 4.1-CNTR #) MOV
5 80 # RDO #) MOV TRUE # TDO-FLG #) MOV
6 RET
7
8 DECIMAL
9
10
11
12
13
14
15

```

```

205
4.1-SEC - Called when RDO transmission ends (4.096 sec) or
4.1 sec timeout .
Start telemetry frame, reset 4.1 sec counter.
Start the RDO telemetry controller, set TDO-FLG true

```

```

56
0 \ Analog to digital converter support - Rnd , ad1 16FEB90CDB \ NMI interrupt support
16FEB90CDB
1 HEX
2 LABEL WRT-RND
3 AX SHR U< IF OFF # AL CMP 0<> IF AX INC THEN THEN
4 WRT-BUFR #) CALL RET
5
6 LABEL GET-AD1-A
7 AD1 1 + #) AL MOV AD1 #) AH MOV
8 30 # AH AND 4 # CL MOV AH CL SHR RET
9
10 LABEL GET-AD1-B
11 AD1 2 + #) AL MOV AD1 #) AH MOV
12 03 # AH AND RET
13
14 DECIMAL
15

```

```

206
WRT-RND - Round 10 bit a/d result to 8 bits and save in DACS
buffer . If data = 255 do not round.
in - AX = a/d data .

GET-AD1-A
Format 10 bit data from A/D 1 channel A .
in - CL is used - lost .
out - AX = a/d data .

GET-AD1-B
Format 10 bit data from A/D 1 channel B .
out - AX = a/d data .

```

Forth 83 Model

57

```

0 \ Analog to digital converter support - Ad1
16FEB90CDB
1 HEX
2 LABEL CVRT-AD1
3 88 # AL OR AL AD1 #) BYTE MOV
4 AD-DELAY do loop
5 00 # AD1 3 + #) BYTE MOV
6 AD-TIME DO LOOP AD1 #) AL MOV 44 # AL AND 44 # AL XOR
7 44 # AL CMP 0=
8 IF
9 GET-AD1-A #) CALL AX CX MOV ( CX = chan a )
10 GET-AD1-B #) CALL
11 ELSE
12 AD1-ERRO # AH MOV SYS-ERR #) CALL
13 EEEE # AX MOV EEEE # CX MOV
14 THEN RET
15

```

58

```

0 \ Analog to digital converter support - Ad2
19JAN88CDB
1 HEX
2
3 LABEL GET-AD2-A
4 AD2 1 + #) AL MOV AD2 #) AH MOV
5 30 # AH AND 4 # CL MOV AH CL SHR RET
6
7 LABEL GET-AD2-B
8 AD2 2 + #) AL MOV AD2 #) AH MOV
9 03 # AH AND RET
10
11 DECIMAL
12
13
14
15

```

59

```

0 \ Analog to digital converter support - Ad2
16FEB90CDB
1 HEX
2 LABEL CVRT-AD2
3 88 # AL OR AL AD2 #) BYTE MOV
4 AD-DELAY do loop
5 00 # AD2 3 + #) BYTE MOV
6 AD-TIME DO LOOP AD2 #) AL MOV 44 # AL AND 44 # AL XOR
7 44 # AL CMP 0=
8 IF
9 GET-AD2-A #) CALL AX CX MOV ( CX = chan a )
10 GET-AD2-B #) CALL
11 ELSE
12 AD2-ERRO # AH MOV SYS-ERR #) CALL
13 EEEE # AX MOV EEEE # CX MOV
14 THEN RET
15 DECIMAL

```

207

1MAR88CDB \ NMI interrupt support

```

CVRT-AD1 - Convert analog input to digital output .
Set enable masks, set addresses
Wait for inputs to settle.
Start conversion .
Wait, get convert flags .
IF flags ok then
    get a/d 1 channel A data get a/d 1 channel B data
        out - CX = a/d 1 channel A 10 bit data .
        out - AX = a/d 1 channel B 10 bit data .
ELSE Post error and pass error ( EEEE hex ) data .

```

208

1654h^A335C \ NMI interrupt support

```

GET-AD2-A
Format 10 bit data from A/D 2 channel A .
    in - CL is used - lost .
    out - AX = a/d data .

GET-AD2-B
Format 10 bit data from A/D 2 channel B .
    out - AX = a/d data .

```

209

1MAR88CDB \ NMI interrupt support

```

CVRT-AD2 - Convert analog input to digital output .
Set enable masks, set addresses
Wait for inputs to settle.
Start conversion .
Wait, get convert flags .
IF flags ok then
    get a/d 2 channel A data get a/d 2 channel B data
        out - CX = a/d 2 channel A 10 bit data .
        out - AX = a/d 2 channel B 10 bit data .
ELSE Post error and pass error ( EEEE hex ) data .

```

Forth 83 Model

```

60
0 \ Langmuir probe support - Lp Ne and Temp
19JAN88CDB
1 HEX
2
3 LABEL GET-LP-NE
4 LP-SWEEP 10 * LP-NE + # AX MOV

5 CVRT-AD2 #) CALL WRT-RND #) CALL
6 LP-CNTR #) AX MOV OA # DL MOV DL DIV 00 # AH CMP 0=
7 IF CX AX MOV WRT-RND #) CALL THEN RET
8
9 LABEL GET-LP-TEMP
10 LP-TEMP # AX MOV CVRT-AD2 #) CALL WRT-RND #) CALL RET
11
12 DECIMAL
13
14
15

```

```

61
0 \ Langmuir probe support - Lp low level control
19JAN88CDB
1 HEX
2
3 LABEL LP-A/D
4 00 # LP-CNTR #) CMP
and
5 0= IF GET-LP-TEMP #) CALL THEN
6 LP-SAMPLES 1+ # LP-CNTR #) CMP
7 U< IF GET-LP-NE #) CALL LP-CNTR #) INC
8 ELSE GET-LP-TEMP #) CALL FALSE # LP-FLG #) MOV THEN
9 RET
10
11 DECIMAL
12
13
14
15

```

```

62
0 \ Met irq interrupt support - Met update data input
11FEB90CDB
1 HEX
2
3 LABEL MET-INT (S -- )
4 01 # MET 3 + #) TEST 0<>
5 IF
6 MET 1 + #) AL MOV MET 2 + #) AH MOV ax m-temp #) mov
7 AH CL MOV 30 # CL AND
8 30 # CL CMP 0=
9 IF 00 # MET-SET 0 + #) MOV OC # AH AND
10 AH SHR AH SHR MET-SET 2 + #) AL MOV 7C # AL AND
11 AL AH ADD AH MET-SET 2 + #) MOV
12 THEN
13
14 DECIMAL
15

```

```

210
2FEB88CDB \ NMI interrupt support

```

GET-LP-NE - Save langmuir probe Ne and Sweep data in the
DACS buffer .
Set a/d addresses for LP sweep and ne data .
Convert data and save ne (rounded byte) in DACS buffer

IF this is the tenth sample save sweep voltage also
in - AX is used - lost .
in - CX is used - lost .
in - DX is used - lost .

GET-LP-TEMP - Save langmuir probe temperature data in the
DACS buffer .

Set a/d addresses for LP temperature and convert data .
Save temperature (rounded byte) in DACS buffer .

in - AX is used - lost .
in - CX is used - lost .
in - DX is used - lost .

```

211
9JUN89CDB \ NMI interrupt support

```

LP-A/D - Langmuir probe control algorithm .
IF this is the first pass save lp temperature .
IF less than LP-SAMPLES have been taken take another sample
increment the sample counter .
ELSE take a lp temperature sample and turn off LP-FLG .

```

212
17SEP89CDB \ Met irq interrupt support - Met update data input

```

LABEL MET-INT (S --)
IF not a sync interrupt THEN
IF this is MET word # 3 format and save in MET-SET .
IF this is MET word # 2 format and save in MET-SET .
IF this is MET word # 1 format and save in MET-SET .
ELSE set sync flag on

Forth 83 Model

63

0 \ Met irq interrupt support - Met update data input 14FEB90CDB \ IRQ interrupt support - Top Level

14SEP89CDB

1 HEX

```

2
3 20 # CL CMP 0=                                IF this is MET word # 2 format and save in MET-SET .
4 IF AL CH MOV AX SHL AX SHL AX SHL 7F # AH AND
5 AH MET-SET 3 + #) MOV MET-SET 2 + #) AH MOV 03 # AH AND
6 7C # CH AND CH AH ADD AH MET-SET 2 + #) MOV
7 THEN
8
9 10 # CL CMP 0=                                IF this is MET word # 1 format and save in MET-SET .
10 IF AL CH MOV AX SHL AX SHL 3F # AH AND
11 AH MET-SET 5 + #) MOV 3F # CH AND CH MET-SET 4 + #) MOV
12 THEN RET
13 ELSE TRUE # MET-SYNC #) MOV THEN RET
14 DECIMAL
15

```

64

0 \ NMI interrupt support - Met clock control 14SEP89CDB \ NMI interrupt support

16FEB90CDB

1 HEX

```

2
3 LABEL INC-MET                                INC-MET - Increment the seconds count of MET clock .
4 MCLK 4 + #) CX MOV MCLK 2 + #) AX MOV AL INC    Move double word to registers and increment the seconds .
5 3C # AL CMP U>= IF 00 # AL MOV AH INC THEN      IF seconds = 60 clear seconds and increment minutes .
6 3C # AH CMP U>= IF 00 # AH MOV CL INC THEN      IF minutes = 60 clear minutes and increment hours .
7 18 # CL CMP U>= IF 00 # CL MOV CH INC THEN      IF hours = 24 clear hours and increment minutes .
8 28 # CH CMP U>= IF 00 # CH MOV AX PUSH MET-ERRO # AX MOV    IF days = 40 clear days and post error .
9 SYS-ERR #) CALL AX POP THEN                    Restore MET clock data .
10 CX MCLK 4 + #) MOV AX MCLK 2 + #) MOV          in - AX is used - lost .
11 RET                                           in - CX is used - lost .
12
13 DECIMAL
14
15

```

65

0 \ NMI interrupt support - Command decoder 16FEB90CDB \ NMI interrupt support - Command decoder

16FEB90CDB

1 HEX

```

2
3 LABEL POST-CMD
4 CSEL-FLG # CMD-PTR #) CMP U>=
5 IF TPM-FLG # CMD-PTR #) CMP 0=
6 IF FF # AL XOR TPM-CNT #) INC
7 AL TPM-CMD #) MOV TPM-CDR #) AL MOV THEN
8 PSEL-FLG # CMD-PTR #) CMP 0=
9 IF AL NEW-PAR #) MOV THEN
10 CSEL-FLG # CMD-PTR #) CMP 0=
11 IF AL NEW-CAL #) MOV THEN
12 CMD-FLG # CMD-PTR #) MOV

If 2nd word of 2 word command then
If tpm-flg then
    incr tpm-cnt, cmd to tpm and cmd-rd to tpm.
If psel-flg then cmd to new-par.
If csel-flg then cmd to new-cal.
Set cmd-flg.

```

Forth 83 Model

66

0 \ NMI interrupt support - Command decoder

16FEB90CDB

1 HEX

2

3 ELSE

4 TPMM-CMD # AL CMP 0=

5 IF TPM-FLG # CMD-PTR #) MOV 00 # CMD-CLK #) MOV THEN

6 PSEL-CMD # AL CMP 0=

7 IF PSEL-FLG # CMD-PTR #) MOV 00 # CMD-CLK #) MOV THEN

8 CSEL-CMD # AL CMP 0=

9 IF CSEL-FLG # CMD-PTR #) MOV 00 # CMD-CLK #) MOV THEN

10 CSEL-FLG # CMD-PTR #) CMP U<

11 IF AX NEW-CMD #) MOV CMD-CNT #) INC

12 CMD-LIMIT # CMD-CLK #) MOV THEN

13 THEN RET

14

15

67

0 \ NMI interrupt support - Background routine 2.5ms nmi18JUL90CDB \ NMI interrupt support - Background routine 2.5ms

nmi16FEB90CDB

1 HEX

2 LABEL NMI-INT (S --)

3 AX PUSH BX PUSH CX PUSH DX PUSH DI PUSH SI PUSH ES PUSH

4 DS PUSH BP PUSH CS AX MOV AX DS MOV

5

6 BONE # CX MOV CL DIG 1 + #) MOV CH DIG 1 + #) MOV

7

8 CMD-LIMIT # CMD-CLK #) CMP U<= IF CMD-CLK #) INC THEN

9

10 PASS-CNT #) INC 01 # PASS-CNT #) BYTE TEST 0<>

11 IF 05 # AX MOV AX PCLK 2+ #) ADD 0 # PCLK #) ADC

12 AX 4.1-CNTR #) ADD

13 TRUE # LP-FLG #) CMP 0= IF LP-A/D #) CALL THEN

14 THEN

15

68

0 \ NMI interrupt support - Cmd check

16FEB90CDB

1 HEX

2 CMD 1+ #) AH MOV AH CMD 1+ #) CMP 0<> IF CMD 1+ #) AH MOV THEN

3 01 # AH TEST 0=

4 IF CMD #) AL MOV OC # AL TEST 0=

5 IF AX PUSH 4 # CL MOV AL CL ROL

6 CMD-LIMIT # CMD-CLK #) CMP U>=

7 IF CMD-FLG # CMD-PTR #) MOV THEN

8 POST-CMD #) CALL AX POP

9 THEN

10 OF # AL TEST 0<>

11 IF CMD-ERRO # AH MOV SYS-ERR #) CALL THEN

12 THEN

13 DECIMAL

14

15

216

16FEB90CDB \ NMI interrupt support - Command decoder

Else

If cmd is tpm-cmd then set tpm-flg and start cmd-clk.

If cmd is psel-cmd then set psel-flg and start cmd-clk.

If cmd is csel-cmd then set csel-flg and start cmd-clk.

If cmd is main-cmd then

post new command, incr cmd-cnt and stop cmd-clk.

ret.

217

0 \ NMI interrupt support - Background routine 2.5ms nmi18JUL90CDB \ NMI interrupt support - Background routine 2.5ms

nmi16FEB90CDB

1 HEX

2 LABEL NMI-INT (S --)

3 AX PUSH BX PUSH CX PUSH DX PUSH DI PUSH SI PUSH ES PUSH

4 DS PUSH BP PUSH CS AX MOV AX DS MOV

5

6 BONE # CX MOV CL DIG 1 + #) MOV CH DIG 1 + #) MOV

7

8 CMD-LIMIT # CMD-CLK #) CMP U<= IF CMD-CLK #) INC THEN

9

10 PASS-CNT #) INC 01 # PASS-CNT #) BYTE TEST 0<>

11 IF 05 # AX MOV AX PCLK 2+ #) ADD 0 # PCLK #) ADC

12 AX 4.1-CNTR #) ADD

13 TRUE # LP-FLG #) CMP 0= IF LP-A/D #) CALL THEN

14 THEN

15

NMI-INT (S --)

Save all cpu registers on the stack.

Move CS to DS.

If cmd-clk <= cmd-limit then incr cmd-clk.

If pass-cnt is odd then.

Add 5 milliseconds to PCLK (pasp mission clock).

Add 5 milliseconds to 4.1-cntr (telemetry scheduler).

If LP-FLG is on take 1 sample of langmuir probe data.

Feed the watchdog timer.

218

16FEB90CDB \ NMI interrupt support - Cmd check

Debounce CMD read.

If new command then

If new command received ok then

If cmd-clk has timed out then set cmd-flg.

Call Post-cmd

If command receive error then post error.

Forth 83 Model


```

69
0 \ NMI interrupt support - TDO check
16FEB90CDB
1 HEX
2
3 TRUE # TDO-FLG #) CMP 0=
4 IF -1 # STEP #) CMP 0=
5 IF 80 # TDO #) BYTE MOV FALSE # TDO-FLG #) MOV
6 4 DO LOOP 00 # TDO #) BYTE MOV
7 THEN
8 THEN
9
10 DECIMAL
11
12
13
14
15

```

```

70
0 \ NMI interrupt support - Service MET
16FEB90CDB
1 HEX
2 MET #) AL MOV AL MET #) CMP 0<> IF MET #) AL MOV THEN
3 AX PUSH FF # AL XOR MET-IMAGE #) AL AND 80 # AL AND 0<>
4 IF 80 # MET-IMAGE #) TEST 0<> IF INC-MET #) CALL THEN THEN
5 AX POP AL MET-IMAGE #) MOV
6 01 # MET-FLG #) TEST 0=
7 IF MET-INT #) CALL 00 # MET-CLR #) BYTE MOV MET-CNT #) INC
8 THEN
9 TRUE # MET-SYNC #) CMP 0=
10 IF FALSE # MET-SYNC #) MOV
11 MET-SET 4 + #) AX MOV AX MCLK 4 + #) MOV
12 MET-SET 2 + #) AX MOV AX MCLK 2 + #) MOV
13 MET-SET 0 + #) AX MOV AX MCLK 0 + #) MOV
14 THEN
15 DECIMAL

```

```

71
0 \ NMI interrupt support - Rdo start and frame check
16FEB90CDB
1 HEX
2 RDO #) AL MOV AL RDO #) CMP 0<> IF RDO #) AL MOV THEN
3 AX PUSH FF # AL XOR RDO-IMAGE #) AL AND 04 # AL AND 0<>
4 IF 4.1-SEC #) CALL
5 ELSE
6 1004 # 4.1-CNTR #) CMP U>=
7 IF 4.1-SEC #) CALL THEN
8 THEN
9 AX POP AL RDO-IMAGE #) MOV
10
11 STEP #) AX MOV 8000 # AX TEST 0<>
transfer
12 IF DACS->TEL #) CALL THEN
13
14 DECIMAL
15

```

```

219
16FEB90CDB \ NMI interrupt support - TDO check

```

IF TDO-FLG is true and it is 320 milliseconds into the frame transfer start the TDO controller and clear TDO-FLG . STEP is set to -1 at end of telemetry schedule.

```

220
16FEB90CDB \ NMI interrupt support - Service MET

```

Debounce Met seconds.
If MET seconds changed then
If from 1 to 0 then incr MCLK.
Save MET-IMAGE
If MET interrupt then
Service interrupt, clr interrupt and incr MET-CNT.
If MET-SYNC is true then
Set MET-SYNC false and move MET-SET to MCLK .

```

221
14JUN89CDB \ NMI interrupt support - Rdo start and frame check

```

Debounce RDO -C9 count.
If C9 count has changed then Start Telemetry frame.
Else If 4.1-cntr has timed out then Start Telemetry frame.
Save RDO-IMAGE .

IF STEP b15 bit is set do 1 step of the telemetry data

Forth 83 Model

```

72
0 \ NMI interrupt support - Over current detector
16FEB90CDB
1 HEX
2
3 DIG #) AL MOV AL DIG #) CMP 0<> IF DIG #) AL MOV THEN
4 AX PUSH FF # AL XOR DIG-IMAGE #) AL AND OC-BIT # AL AND 0<>
5 IF OC-BIT # DIG-IMAGE #) TEST 0<>
6 IF OC1-CNT #) INC THEN THEN
7 AX POP AL DIG-IMAGE #) MOV
8
9 00 # TMR-CLR #) BYTE MOV ( Timing pulse )
10
11 BP POP DS POP
12 ES POP SI POP DI POP DX POP CX POP BX POP AX POP
13 IRET
14
15

```

```

73
0 \ Telemetry packet support - Telemetry overhead table
16FEB90CDB
1 HEX
2
3 CODE GET-TBL (S offset,addr -- addr')
4 AX POP CX POP BX PUSH AX BX MOV CS: 0 [BX] DI MOV
5 CX BX MOV CS: 0 [BX+DI] AX MOV BX POP 1PUSH C;
6
7 ALSO ASSEMBLER
8 HERE ' TEL-TBL >BODY THERE !
9 TEL-00 , TEL-01 , TEL-02 , TEL-03 , TEL-04 ,
10 TEL-05 , TEL-06 , TEL-07 , TEL-08 , TEL-09 ,
11 TEL-10 , TEL-11 , TEL-12 , TEL-13 ,
12
13 DECIMAL
14
15

```

```

74
0 \ PROTECTED MEMORY ACCESS
16SEP89CDB
1 HEX
2
3 : PMEM-WAIT (S addr -- )
4 P-ADDR ! P-FLG ON BEGIN P-FLG @ FALSE = UNTIL ;
5
6 : PMEM! (S data,addr -- flg)
7 P-WRT ON SWAP P-DATA ! PMEM-WAIT P-ADDR @ ;
8
9 : PMEM@ (S addr -- data,flg)
10 P-WRT OFF PMEM-WAIT P-DATA @ P-ADDR @ ;
11
12 DECIMAL
13
14
15

```

```

222
15JUL90CDB \ NMI interrupt support - Over current detector
HEX
Debounce dig read.
If oc-bit has changed then
If oc-bit is 0 then
increment the hv-oc-cnt.
Save dig image.
Clear p-mem gate. Scope on SEQ board to observe nmi timing.
Restore registers and return from interrupt.

```

```

223
15JUN90CDB \ NMI interrupt support - Over current detector
GET-TBL (S offset,addr -- addr')
Registers Lost - AX,CX
TEL-TBL - Telemetry overhead table.
TEL-00 , TEL-01 , TEL-02 , TEL-03 , TEL-04 ,
TEL-05 , TEL-06 , TEL-07 , TEL-08 , TEL-09 ,
TEL-10 , TEL-11 , TEL-12 , TEL-13 ,

```

```

224
16SEP89CDB \ PROTECTED MEMORY ACCESS
PMEM! (S data,addr -- )
- Save data to protected ram 0 <= addr <= 63 data is a byte
PMEM@ (S addr -- data)
- Read data from protected ram 0 <= addr <= 63 data is a

```

Forth 83 Model

75

0 \ DACS DIGITAL I/O BOARD SUPPORT

16FEB90CDB

1 HEX

2 LABEL (DIG-EN)

3 4 DO CX BX MOV OFF # PM [BX] BYTE MOV LOOP

4 OFF # DIG 3 + #) BYTE MOV RET

5

6 code d-en OFF # DIG 3 + #) BYTE MOV next c;

7 CODE DIG-EN (S --)

8 (DIG-EN) #) CALL NEXT C;

9

10 : DIG! (S data,addr --) [DIG 4 +] LITERAL + C! d-en ;

11
12 : DIG@ (S addr -- data) [DIG] LITERAL + C@ ;

13

14 DECIMAL

15

76

0 \ DACS DIGITAL I/O BOARD CHECKOUT SUPPORT

16FEB90CDB

1 HEX

2

3 : HV-STAT? (S -- stat)

4 0 DIG@ 4 / 3 AND ;

5

6 DECIMAL

7

8

9

10

11

12

13

14

15

77

0 \ Analog to digital converter support - Forth words

16SEP89CDB

1 HEX

2

3 CODE AD1-CVRT (S ab-addr -- b-data,a-data)

4 AX POP CVRT-AD1 #) CALL AX PUSH CX PUSH NEXT C;

5

6 CODE AD2-CVRT (S ab-addr -- b-data,a-data)

7 AX POP CVRT-AD2 #) CALL AX PUSH CX PUSH NEXT C;

8

9 DECIMAL

10

11

12

13

14

15

225

17JUL90CDB \ DACS DIGITAL I/O BOARD SUPPORT

LABEL (DIG-EN)

Registers Lost - AX,CX

DIG-EN (S --)

- Enable tri-state outputs on digital interface board .

DIG! (S data,addr --)

- Save data to digital board 0 <= addr <= 3 data is a byte

DIG@ (S addr -- data)

- Read data from digital board 0 <= addr <= 3 data is a byte

226

11FEB90CDB \ DACS DIGITAL I/O BOARD CHECKOUT SUPPORT

HV-STAT? (S -- stat)

- Read hv polarity and over current flag .

227

16FEB90CDB \ Analog to digital converter support - Forth words

AD1-CVRT (S ab-addr -- b-data,a-data)

- A Forth version of the a/d convert routine for a/d 1 .
(ab-addr is a byte of the form a-addr * 16 + b-addr)

AD2-CVRT (S ab-addr -- b-data,a-data)

- A Forth version of the a/d convert routine for a/d 2 .
(ab-addr is a byte of the form a-addr * 16 + b-addr)

Forth 83 Model

```

78
0 \ DACS RTD BOARD SUPPORT
16FEB90CDB
1 HEX
2
3 CODE RTD-CVRT (S chan -- data)
4 AX POP 07 # AL AND 4 # CL MOV AL CL SHL AL RTD #) MOV
5 00 # RTD 1 + #) BYTE MOV
6 AD-TIME DO LOOP RTD #) AL MOV 40 # AL AND 40 # AL XOR
7 40 # AL CMP 0=
8 IF
9 RTD 1 + #) AL MOV RTD #) AH MOV
10 30 # AH AND 4 # CL MOV AH CL SHR
11 ELSE
12 RTD-ERR0 # AH MOV SYS-ERR #) CALL EEEE # AX MOV
13 THEN AX PUSH NEXT C;
14 DECIMAL
15

```

```

79
0 \ System timing words
16FEB90CDB
1 DECIMAL
2
3 : MILLISECONDS (S ms -- )
4 10 25 */
5 1 MAX PASS-CNT @ +
6 BEGIN DUP PASS-CNT @ = UNTIL DROP ;
7
8 : SECONDS (S n -- )
9 0 ?DO 1000 MILLISECONDS LOOP ;
10
11 : MINUTES (S n -- )
12 0 ?DO 60 SECONDS LOOP ;
13
14
15

```

```

80
0 \ PASP TELEMETRY BUFFER SUPPORT
16SEP89CDB
1 HEX
2 CODE T-BUF-INIT (S -- ) INIT-BUF #) CALL NEXT C;
3
4 CODE WRT-BUF (S word -- )
5 AX POP AX PUSH WRT-BUFR #) CALL
6 AX POP AH AL MOV WRT-BUFR #) CALL NEXT C;
7
8 CODE E-BUF-INIT (S -- ) INIT-E-BUF #) CALL NEXT C;
9
10 CODE ERROR! (S err# -- ) AX POP SYS-ERR #) CALL NEXT C;
11
12 CODE ERROR? (S -- code/false )
13 RD-E-BUF #) CALL AX PUSH NEXT C;
14 DECIMAL
15

```

```

228
1654h^A335C \ DACS RTD BOARD SUPPORT

```

```

RTD-CVRT - Convert analog input to digital output .
Set enable masks, set addresses
Start conversion .
Wait, get convert flags .
IF flags ok then Get RTD data.
ELSE Post error and pass error ( EEEE hex ) data .

```

```

229
2MAY90CDB \ IV SUPPORT

```

```

MILLISECONDS (S ms -- )
- Wait for ms Milliseconds .

SECONDS (S n -- )
- Wait for n Seconds .

MINUTES (S n -- )
- Wait for n MINUTES .

```

```

230
16SEP89CDB \ PASP TELEMETRY BUFFER SUPPORT

```

```

T-BUF-INIT (S -- ) - forth version of INIT-BUF

WRT-BUF (S word -- ) - Send word to the dacs buffer .

E-BUF-INIT (S -- ) - forth version of INIT-E-BUF

ERROR! (S err# -- ) - Forth version of SYS-ERR .

ERROR? (S -- code/false ) - Forth version of RD-E-BUF .

```

Forth 83 Model

81

0 \ PASP TELEMETRY BUFFER SUPPORT

19JAN88CDB

1 HEX

2

3 CODE T-BUF-C! (S byte --)

4 AX POP WRT-BUFR #) CALL NEXT C;

5

6 : T-BUF-! (S n --)

7 WRT-BUF ;

8

9 CODE T-BUF-CS! (S n --)

10 AX POP WRT-RND #) CALL NEXT C;

11

12 DECIMAL

13

14

15

82

0 \ PASP TELEMETRY BUFFER - Marker support

16FEB90CDB

1 HEX

2 : L-NAME (S str,len --)

3 2DUP + -ROT DROP DO I C@ T-BUF-C! LOOP ;

4

5 : CNT+! (S addr --) 1 OVER +! C@ T-BUF-C!

6 PCLK 2@ SWAP T-BUF-! T-BUF-! ;

7

8 : (CAL-PAR) (S p1,p2 --) T-BUF-C! T-BUF-C! ;

9

10 T: .RP (S --) [TARGET] NOOP T;

11

12 : L-TM (S --) "TIM" L-NAME TIM-CNT CNT+! ;

13 DECIMAL

14

15

83

0 \ PASP TELEMETRY BUFFER SUPPORT - EVENT MARKERS

16FEB90CDB

1 HEX

2

3 : L-RES (S --)

4 "RES" L-NAME RES-CNT CNT+! ;

5

6 : L-DOG (S --)

7 "WDG" L-NAME WDG-CNT CNT+! ;

8

9 : L-ORBIT (S n --)

10 "OR" L-NAME 30 + T-BUF-C! ORB-CNT CNT+! ;

11

12 : L-SEG (S n --)

13 "SE" L-NAME 30 + T-BUF-C! SEG-CNT CNT+! ;

14 DECIMAL

15

231

1AUG89CDB \ PASP TELEMETRY BUFFER SUPPORT

T-BUF-C! (S byte --)

- Save 1 byte in DACS buffer . A forth version of WRT-BUFR

T-BUF-! (S n --)

- Save 1 word in DACS buffer . A synonym for WRT-BUF .

T-BUF-CS! (S n --)

- Save 1 rounded byte in DACS buffer .

A forth version of WRT-RND .

232

18JUL90CDB \ PASP TELEMETRY BUFFER SUPPORT - BUFFER LINK SUPPORT

: L-NAME (S str,len --)

- Store a name in the DACS buffer .

CNT+! (S addr --)

- Save current count to buffer then increment count.

(CAL-PAR) (S p1,p2 --)

- Save 2 bytes to Data buffer.

233

18JUL90CDB \ PASP TELEMETRY BUFFER SUPPORT - BUFFER LINK SUPPORT

L-RESET (S --)

- Make a "RES" marker (4 bytes).

L-W-DOG (S --)

- Make a "WDG" marker (4 bytes).

L-ORBIT (S n --)

- Make an "ORn" marker and append the orbit number

1 <= n <= 6 (4 bytes).

L-SEG (S n --)

- Make a "SEn" marker and append the segment number

1 <= n <= 3 (4 bytes).

Forth 83 Model

84

```

0 \ PASP TELEMETRY BUFFER SUPPORT - EVENT MARKERS
16FEB90CDB
1 HEX
2 : L-EL (S n -- )
3   " EL" L-NAME 30 + T-BUF-CI ELE-CNT CNT+1 ;
4 : L-STBY (S -- )
5   " SBY" L-NAME SBY-CNT CNT+1 ;
6 : L-GARR (S -- )
7   " GAR" L-NAME GAR-CNT CNT+1 ;
8 : L-OCC (S -- )
9   " OCC" L-NAME OCC-CNT CNT+1 ;
10 : L-HK (S -- )
11   " HSK" L-NAME HSK-CNT CNT+1 ;
12 : L-IV (S n -- )
13   " IV" L-NAME 30 + T-BUF-CI IVS-CNT CNT+1 ;
14 DECIMAL
15

```

85

```

0 \ PASP TELEMETRY BUFFER SUPPORT - DATA MARKERS
16FEB90CDB
1 HEX
2
3 : L-TP (S -- )
4   " TPM" L-NAME TPM-CNT CNT+1 ;
5
6 : L-LP (S -- )
7   " LMP" L-NAME LMP-CNT CNT+1 ;
8
9 : L-HV (S -- )
10   " HVS" L-NAME HVS-CNT CNT+1 ;
11
12 : L-HV2 (S -- )
13   " HV2" L-NAME HVS-CNT CNT+1 ;
14 DECIMAL
15

```

86

```

0 \ PASP TELEMETRY BUFFER SUPPORT - DATA MARKERS
16FEB90CDB
1 HEX
2 : L-TSS (S -- ) " TSS" L-NAME TSS-CNT CNT+1 ;
3
4 : L-TST (S -- ) " TST" L-NAME TST-CNT CNT+1 ;
5
6 : L-CA (S -- ) " CAL" L-NAME CAL-CNT CNT+1 ;
7
8 : L-SS (S -- ) " SSI" L-NAME CAL-CNT CNT+1 ;
9
10 : L-PG (S -- ) " PGI" L-NAME CAL-CNT CNT+1 ;
11
12 : L-RESET (S -- )
13   L-RES RDO-STAMP @ T-BUF-1 CMD-STAMP @ T-BUF-1 ;
14 DECIMAL
15

```

234

```

15AUG89CDB \ PASP TELEMETRY BUFFER SUPPORT - BUFFER LINK SUPPORT

L-EL (S n -- )
- Make a " ELn" marker and append the element number
1 <= n <= 4 ( 4 bytes ).
L-STBY (S -- )
- Make a " SBY" marker ( 4 bytes ).
L-GARR (S -- )
- Make a " GAR" marker ( 4 bytes ).
L-OCC (S -- )
- Make a " OCC" marker ( 4 bytes ).
L-HK (S -- )
- Make a " HSK" marker ( 4 bytes ).
L-IV (S n -- )
- Make a " IVn" marker and append the array number
1 <= n <= 6 ( 4 bytes ).

```

235

```

3APR90CDB \ PASP TELEMETRY BUFFER SUPPORT - BUFFER LINK SUPPORT

T-LP (S -- )
- Make a " TPM" marker ( 4 bytes ).

L-LP (S -- )
- Make a " LMP" marker ( 4 bytes ).

L-HV (S volts,cmd -- )
- Make a " HVS" marker .
volts is a byte . cmd is a byte ( 4 bytes ).

L-HV2(S volts,cmd -- )
- Make a " HV2" marker .
volts is a byte . cmd is a byte ( 4 bytes ).

```

236

```

18JUL90CDB \ PASP TELEMETRY BUFFER SUPPORT - BUFFER LINK SUPPORT

L-TSS (S -- )
- Make a " TSS" marker ( 4 bytes ).

L-TST (S -- )
- Make a " TST" marker ( 4 bytes ).

L-CA (S -- )
- Make a " CAL" marker ( 4 bytes ).

L-SS (S -- ) " SSI" marker ( 4 bytes ).

L-PG (S -- ) " PGI" marker ( 4 bytes ).

```

Forth 83 Model

87

```

0 \ PASP HOUSEKEEPING SUPPORT
19JAN88CDB
1 HEX
2
3 : PM->BUF (S -- )
4 [ PM 1 + ] LITERAL C@ T-BUF-C!
5 [ PM 2 + ] LITERAL C@ T-BUF-C! ;
6
7 : TEMP-PRESS->BUF (S -- )
8 [ PRESS2 10 * TEMP2-1 + ] LITERAL AD1-CVRT
9 SWAP T-BUF-CS! T-BUF-CS!
10 [ PRESS1 10 * TEMP1-1 + ] LITERAL AD2-CVRT
11 SWAP T-BUF-CS! T-BUF-CS! ;
12
13 DECIMAL
14
15

```

88

```

0 \ PASP HOUSEKEEPING SUPPORT
16FEB90CDB
1 HEX
2
3 : AUX-TEMPS->BUF (S -- )
4 [ TEMP1-2 10 * TEMP2-2 + ] LITERAL AD2-CVRT
5 SWAP T-BUF-CS! T-BUF-CS! ;
6
7 : ERROR->BUF (S -- )
8 E-BUF-GAUGE @ T-BUF-C!
9 [ E-BUF-SIZE ] LITERAL 2/ 0 DO ERROR? T-BUF-! LOOP ;
10
11 : HK-HEALTH (S -- )
12 PM->BUF TEMP-PRESS->BUF AUX-TEMPS->BUF ERROR->BUF ;
bytes
13
14 DECIMAL
15

```

89

```

0 \ PASP HOUSEKEEPING SUPPORT
19JAN88CDB
1 HEX
2 : A/D-CAL->BUF (S -- )
3 [ CAL1A 10 * CAL1B + ] LITERAL AD1-CVRT T-BUF-! T-BUF-!
4 [ CAL2A 10 * CAL2B + ] LITERAL AD2-CVRT T-BUF-! T-BUF-! ;
5
6 : RTD-CAL->BUF (S -- )
7 [ RTD-CAL1 ] LITERAL RTD-CVRT T-BUF-!
8 [ RTD-CAL2 ] LITERAL RTD-CVRT T-BUF-! ;
9
10 : MET->BUF (S -- )
11 PCLK 2@ SWAP T-BUF-! T-BUF-! MCLK 2 + 2@ SWAP T-BUF-! T-BUF-!
12 MCLK @ T-BUF-! ;
13 DECIMAL
14
15

```

237

11FEB90CDB \ PASP HOUSEKEEPING SUPPORT

```

PM->BUF (S -- )
- Write power monitor data to DACS buffer . 2 bytes .

TEMP-PRESS->BUF (S -- )
- Write box pressure and temperature for boxes 1 and 2 to
  buffer . 4 rounded bytes .

```

238

21MAR90CDB \ PASP HOUSEKEEPING SUPPORT

```

AUX-TEMPS->BUF (S -- )
- Write aux temperatures data to DACS buffer . 2 rounded bytes

ERROR->BUF (S -- )
- Dump error buffer . 33 bytes .

HK-HEALTH (S -- )
PM->BUF TEMP-PRESS->BUF AUX-TEMPS->BUF ERROR->BUF
  2         4             2             33 = 41

```

239

18JUL90CDB \ PASP HOUSEKEEPING SUPPORT

```

A/D-CAL->BUF (S -- )
- Write a/d calibration words data to DACS buffer . 8 bytes

RTD-CAL->BUF (S -- )
- Write rtd calibration words data to DACS buffer . 4 bytes

MET->BUF (S -- )
- Write pasp clock and MET clock to DACS buffer . 10 bytes

```

Forth 83 Model

90

0 \ PASP HOUSEKEEPING SUPPORT

16FEB90CDB

1 HEX

2

3 : HK-STATUS->BUF (S --)

4 RES-CNT @ T-BUF-C! WDG-CNT @ T-BUF-C! JRB-CNT @ T-BUF-C!

5 SEG-CNT @ T-BUF-C! ELE-CNT @ T-BUF-C! SBY-CNT @ T-BUF-C!

6 GAR-CNT @ T-BUF-C! OCC-CNT @ T-BUF-C! TSS-CNT @ T-BUF-C!

7 TST-CNT @ T-BUF-C! TIM-CNT @ T-BUF-C!

8 HSK-CNT @ T-BUF-C! IVS-CNT @ T-BUF-C! HVS-CNT @ T-BUF-C!

9 TPM-CNT @ T-BUF-C! LMP-CNT @ T-BUF-C! CAL-CNT @ T-BUF-C!

10 CMD-CNT @ T-BUF-C! INT-CNT @ T-BUF-C! TPM-CNT @ T-BUF-C!

11 MET-CNT @ T-BUF-C! OCI-CNT @ T-BUF-C!

12 NEW-FAR @ T-BUF-C! NEW-CAL @ T-BUF-C! NEW-CMD @ T-BUF-C!

13 SP@ T-BUF-I RP@ @ T-BUF-I ;

14 DECIMAL

15

240

18JUL90CDB \ PASP HOUSEKEEPING SUPPORT

HK-STATUS->BUF (S --)

- Write status data to DACS buffer . 29 bytes .

91

0 \ PASP HOUSEKEEPING SUPPORT - SYSTEM STAT

16FEB90CDB

1 HEX

2

3 : HK-CAL (S --)

4 A/D-CAL->BUF RTD-CAL->BUF MET->BUF ;

5

6 : PG->BUF (S --)

7 [PG-PRESS 10 * PG-TEMP +] LITERAL AD1-CVRT

8 T-BUF-CS! T-BUF-CS! ;

9

10 : (PG->BUF) (S --) L-PG PG->BUF ;

11

12

13

14

15

241

12FEB90CDB \ PASP HOUSEKEEPING SUPPORT - SYSTEM STATUS

HK-CAL (S --)

A/D-CAL->BUF RTD-CAL->BUF MET->BUF ;
8 4 10 = 22 bytes

PG-PRESS - 2 bytes.

92

0 \ PASP HOUSEKEEPING SUPPORT - SYSTEM STATUS

16FEB90CDB

1 HEX

2

3

4 : HK (S --)

5

6 .RP L-HK HK-HEALTH HK-CAL HK-STATUS->BUF PG->BUF ;

7 DECIMAL

8

9

10

11

12

13

14

15

242

16FEB90CDB \ PASP HOUSEKEEPING SUPPORT - SYSTEM STATUS

HK (S --)

- Make a lp header and write housekeeping data to DACS buffer

L-HK HK-HEALTH HK-CAL HK-STATUS->BUF PG->BUF .
4 41 22 29 2 = 98 bytes

Forth 83 Model

<pre> 93 0 \ IV SUPPORT 1 HEX 2 : ARRAY-TEMP->BUF (S array -- array) 3 DUP 1 - RTD-CVRT T-BUF-! ; 4 5 : SS->BUF (S --) 6 [SSX] LITERAL C@ T-BUF-C! [SSY] LITERAL C@ T-BUF-C! 7 [SSA] LITERAL C@ T-BUF-C! ; 8 9 : (SS->BUF) (S --) L-SS SS->BUF ; 10 11 : R-SETTLE (S --) 10 MILLISECONDS ; 12 13 : IV! (S iv-rset --) FF XOR 0 DIG! ; 14 DECIMAL 15 94 0 \ IV SUPPORT 1 HEX 2 3 : IV-ON (S array --) 4 [IV-SEL] LITERAL + FF XOR 1 DIG! ; 5 6 : IV-OFF (S --) 7 FF 0 DIG! FF 1 DIG! ; 8 9 : IV->BUF (S array,res# -- array) 10 T-BUF-C! DUP DUP 11 1 3 BETWEEN 12 IF 3 + 11 * AD2-CVRT ELSE 11 * AD1-CVRT THEN 13 T-BUF-CS! T-BUF-CS! ; 14 DECIMAL 15 95 0 \ IV SUPPORT 1 HEX 2 3 CODE CS:C@ (S addr --byte) 4 AX POP AX DI MOV CS: 0 [DI] AL MOV 00 # AH MOV 1PUSH C; 5 6 : (IV) (S addr --) 7 DUP CS:C@ DUP IV-ON DUP L-IV SS->BUF 8 ARRAY-TEMP->BUF SWAP 9 1 + DUP 20 + SWAP DO I CS:C@ DUP IV! R-SETTLE IV->BUF LOOP 10 ARRAY-TEMP->BUF 11 IV-OFF 12 DROP ; 13 14 DECIMAL 15 </pre>	<pre> 243 16MAY90CDB \ IV SUPPORT 11FEB90CDB ARRAY-TEMP->BUF (S array -- array) - Write requested array temperature data to DACS buffer . 2 bytes . SS->BUF (S --) - Write sun sensor data to DACS buffer . 3 bytes . R-SETTLE (S --) - Relay settling time (TBD) . IV! (S iv-rset --) - Set resistor select on digital board . 0 <= iv-rset <= 255 244 16MAY90CDB \ IV SUPPORT 19JAN88CDB IV-ON (S array --) - Select requested array for iv measurement . IV-OFF (S --) - De-select arrays turn off iv measurement . IV->BUF (S array,res# -- array) - Write resistor select number, V and I data to DACS buffer 3 bytes . 245 18SEP89CDB \ IV SUPPORT 16FEB90CDB CODE CS:C@ (S addr --byte) Registers Lost - AX,DI - Used to read rom tables . (IV) (S addr --) - Write iv header, sun sensor data, array temperature, 4 3 2 32 iv data packets and the array temperature to DACS buffer 32 * 3 2 = 107 bytes </pre>
--	---

Forth 83 Model

```

96
0 \ IV SUPPORT - IV TABLES 1 - 3
16FEB90CDB
1 HEX
2 ALSO FORTH
3 LABEL IV1-T IV1-RES
4 : IV1 (S -- ) .RP [ ALSO ASSEMBLER IV1-T ] LITERAL (IV) ;
5
6 LABEL IV2-T IV2-RES
7 : IV2 (S -- ) .RP [ ALSO ASSEMBLER IV2-T ] LITERAL (IV) ;
8
9 LABEL IV3-T IV3-RES
10 : IV3 (S -- ) .RP [ ALSO ASSEMBLER IV3-T ] LITERAL (IV) ;
11
12 DECIMAL
13
14
15

```

```

97
0 \ IV SUPPORT - IV TABLES 4 - 6
16FEB90CDB
1 HEX
2
3 LABEL IV4-T IV4-RES
4 : IV4 (S -- ) .RP [ ALSO ASSEMBLER IV4-T ] LITERAL (IV) ;
5
6 LABEL IV5-T IV5-RES
7 : IV5 (S -- ) .RP [ ALSO ASSEMBLER IV5-T ] LITERAL (IV) ;
8
9 LABEL IV6-T IV6-RES
10 : IV6 (S -- ) .RP [ ALSO ASSEMBLER IV6-T ] LITERAL (IV) ;
11
12 DECIMAL
13
14
15

```

```

98
0 \ IV SUPPORT
1 HEX
2
3 : LP-ON (S -- )
4 [ LP-ON-CMD FF XOR ] LITERAL 2 DIGI FF 2 DIGI ;
5
6 : (LP) (S -- )
7 LP-CNTR OFF LP-ON
8 LP-FLG ON BEGIN LP-FLG @ NOT UNTIL ;
9
10 : LP (S -- )
11 .RP L-LP (LP) ;
12
13 DECIMAL
14
15

```

```

246
\ IV SUPPORT - IV TABLES 1 - 3

```

```

247
10AUG89CDB \ IV SUPPORT - IV TABLES 4 - 6

```

```

248
9MAR90CDB \ IV SUPPORT
16FEB90CDB

LP-ON (S -- )
- Send pulse to langmuir probe .

(LP) (S -- )
- Reset lp sample counter, start lp sweep,
start background sample taking and wait till done .
400 + 40 + 2 = 442 bytes

LP (S -- )
- Write lp header do (LP).
4 + 442 = 446 bytes.

```

Forth 83 Model

```

99
0 \ TPM INTERFACE BOARD CHECK OUT
16FEB90CDB
1 HEX
2
3 : INIT-TPM (S -- )
4 OFF [ TPM-FMB ] LITERAL C! 19 MILLISECONDS
request
5 00 [ TPM-ADDR ] LITERAL C! [ TPM-DATA ] LITERAL C@
6 OFF XOR 01 AND 01 = NOT
7 IF OFF [ TPM-FMB ] LITERAL C! THEN ;
8
9 : READ-TPM (S -- )
10 [ TPM-STAT ] LITERAL C@ OFF XOR T-BUF-C!
11 16 0 DO 1 [ TPM-ADDR ] LITERAL C!
12 [ TPM-DATA ] LITERAL C@ OFF XOR T-BUF-C! LOOP ;
13
14 DECIMAL
15

```

```

100
0 \ TPM INTERFAC. BOARD CHECK OUT
16FEB90CDB
1 HEX
2
3 : TPM->BUF (S -- )
4 L-TP OFF [ TPM-FMB ] LITERAL C! 1F4 MILLISECONDS READ-TPM
5 OFF [ TPM-FMB ] LITERAL C! 1F4 MILLISECONDS READ-TPM ;
6
7 DECIMAL
8
9
10
11
12
13
14
15

```

```

101
0 \ HV SUPPORT
1 HEX
2
3 : HV->BUF (S -- )
4 [ ETELEM 10 * ITELEM + ] LITERAL AD1-CVRT
5 T-BUF-CS! T-BUF-CS!
6 [ ELECT-I 10 * EMIT-I + ] LITERAL AD1-CVRT
7 T-BUF-CS! E-TEMP ! HV-STAT? T-BUF-C! ;
8
9 : HV-ON (S -- )
10 [ +000V FF XOR ] LITERAL 2 DIGI
11 [ FPAVE-GND FF XOR ] LITERAL 1 DIGI 32 MILLISECONDS ;
12
13 DECIMAL
14
15

```

```

249
18JUL90CDB \ TPM INTERFACE BOARD CHECK OUT
HEX
INIT-TPM (S -- )
- Initialize TPM to send 1st half of data buffer on next

READ-TPM (S -- )
- Read TPI status byte and 22 bytes of TPI buffer to the
data buffer. 23 bytes.
DECIMAL

```

```

250
21MAR90CDB \ TPM INTERFACE BOARD CHECK OUT
HEX
: TPM->BUF (S -- )
- Send TPM header and the TPM data.
4 + 46 = 50 bytes.
DECIMAL

```

```

251
4MAY90CDB \ HV SUPPORT
16FEB90CDB
HV->BUF (S -- )
- Wait for 2 second then write hv voltage monitor, hv current
monitor, electrometer log 1 and hv status byte . 4 bytes

```

Forth 83 Model

102

0 \ HV SUPPORT

1 HEX

2

3 : HV-OFF (S --)

4 [+000V FF XOR] LITERAL 2 DIG! R-SETTLE

settle

5 [00-SEL FF XOR] LITERAL 1 DIG! R-SETTLE ;

6

7 : HV (S n,volts,cmd --)

8 OVER [-V] LITERAL AND

9 IF [-V] LITERAL OR

10 SWAP [EMIT-ON -V OR NOT] LITERAL AND SWAP THEN

11 2DUP L-HV (CAL-PAR)

+ 7

12 SWAP 1F AND FF XOR 2 DIG! R-SETTLE FF XOR 1 DIG! R-SETTLE

13 0 DO HV->BUF TPM->BUF LOOP E-TEMP @ T-BUF-CS! ;

14 DECIMAL

15

103

0 \ DACS High Voltage Commands

19JAN88CDB

1 HEX

2

3 HERE ' HV-TBL >BODY THERE !

4 (HV-TBL)

5

6 HERE ' HV-C-TBL >BODY THERE !

7 (HV-C-TBL)

8

9 DECIMAL

10

11

12

13

14

15

104

0 \ PASP Calibration support

16FEB90CDB

1 HEX

2

3 : CAL-TD (S --) [CAL-TC] LITERAL MILLISECONDS ;

4

5 : AD1-A-CAL (S --)

6 .RP NEW-PAR C@ 1A OVER L-CA (CAL-PAR) 10 *

7 64 0 DO CAL-TD DUP AD1-CVRT NIP T-BUF-! LOOP DROP ;

8

9 : AD1-B-CAL (S --)

10 .RP NEW-PAR C@ 1B OVER L-CA (CAL-PAR)

11 64 0 DO CAL-TD DUP AD1-CVRT DROP T-BUF-! LOOP DROP ;

12

13 DECIMAL

14

15

252

15JUN90CDB \ HV SUPPORT

16FEB90CDB

HV-OFF (S --)

- De-select array and hv measurement, wait for relays to

and set hv to 0 .

HV (S n,volts,cmd --)

- Make hv header, save cmd, volts, set command, wait for

$$n * (4 + 50) + 1 = n * 54$$

n = 20 for (ORBIT-SEQ) and TEST-SEQ = 1087 bytes.

n = 02 for TEST-SEQ-S (Short test sequence) = 115 bytes.

253

17FEB90CDB \ DACS High Voltage Commands

HV-TBL - High voltage setting codes

+000V , +50V , +100V , +150V , +300V , +500V , -000V

-150V , -200V , -300V , -500V ,

HV-C-TBL - Commands for mission array selection and

configurations .

OR1-EL1-CMD , OR1-EL2-CMD , OR1-EL3-CMD , OR1-EL4-CMD

OR2-EL1-CMD , OR2-EL2-CMD , OR2-EL3-CMD , OR2-EL4-CMD

OR3-EL1-CMD , OR3-EL2-CMD , OR3-EL3-CMD , OR3-EL4-CMD

OR4-EL1-CMD , OR4-EL2-CMD , OR4-EL3-CMD , OR4-EL4-CMD

OR5-EL1-CMD , OR5-EL2-CMD , OR5-EL3-CMD , OR5-EL4-CMD

OR6-EL1-CMD , OR6-EL2-CMD , OR6-EL3-CMD , OR6-EL4-CMD

254

16FEB90CDB \ PASP Calibration support

CAL-TD (S --)

- Time between samples.

AD1-A-CAL (S --)

- Make cal marker with AD1-A (1A) code and channel #

$$2 * 100 = 206 \text{ bytes.}$$

AD1-B-CAL (S --)

- Make cal marker with AD1-B (1B) code and channel #

$$2 * 100 = 206 \text{ bytes.}$$

Forth 83 Model

```

105
0 \ PASP Calibration support
16FEB90CDB
1 HEX
2
3 : AD2-A-CAL (S -- )
4 .RP NEW-PAR C@ 2A OVER L-CA (CAL-PAR) 10 *
5 64 0 DO CAL-TD DUP AD2-CVRT NIP T-BUF-! LOOP DROP ;
6
7 : AD2-B-CAL (S -- )
8 .RP NEW-PAR C@ 2B OVER L-CA (CAL-PAR)
9 64 0 DO CAL-TD DUP AD2-CVRT DROP T-BUF-! LOOP DROP ;
10
11 DECIMAL
12
13
14
15

```

```

106
0 \ PASP Calibration support
16FEB90CDB
1 HEX
2
3 : RTD-CAL (S -- )
4 .RP NEW-PAR C@ 00 OVER L-CA (CAL-PAR)
5 64 0 DO CAL-TD DUP RTD-CVRT T-BUF-! LOOP DROP ;
6
7
8 DECIMAL
9
10
11
12
13
14
15

```

```

107
0 \ Pasp test sequence
16FEB90CDB
1 HEX
2
3 : TEST-SEG (S -- )
4 L-TST
5 HK IV1 IV2 IV3 IV4 IV5 IV6 LP
6 HV-ON
7 9 0 DO
8 0B 0 DO .RP 14 1 2* [ ' HV-TBL >BODY ] LITERAL GET-TBL
9 [ 06 08 * ] LITERAL J 2* + [ ' HV-C-TBL >BODY ] LITERAL
10 GET-TBL HV
11 LOOP HV-OFF
12 LOOP ;
13
14 DECIMAL
15

```

```

255
16FEB90CDB \ PASP Calibration support

```

```

AD2-A-CAL (S -- )
- Make cal marker with AD2-A (2A) code and channel #
  4 + 2
- Read channel pointed to by NEW-PAR 100 times .
  2 * 100 = 206 bytes.

AD2-B-CAL (S -- )
- Make cal marker with AD2-B (2B) code and channel #
  4 + 2
- Read channel pointed to by NEW-PAR 100 times .
  2 * 100 = 206 bytes.

```

```

256
16FEB90CDB \ PASP Calibration support

```

```

RTD-CAL (S -- )
- Make cal marker with RTD (00) code and channel #
  4 + 2
- Read channel pointed to by NEW-PAR 100 times .
  2 * 100 = 206 bytes.

```

```

257
4MAY90CDB \ Pasp test sequence

```

```

HEX
TEST-SEG (S -- )
- Test sequence used to exercise all functions at least once.
  4 + 98 + (6 * 107) + 446 = 1190 bytes
  11 * 9 * (1087) = 107,613 bytes.
total = 108,803 bytes
DECIMAL

```

Forth 83 Model

108

0 \ Pasp short test sequence

16FEB90CDB

1 HEX

2

3 : TEST-SEG-S (S --)

4 L-TSS

5 HK IV1 IV2 IV3 IV4 IV5 IV6 LP

6 HV-ON

7 9 0 DO

8 0B 0 DO .RP 02 I 2* [' HV-TBL >BODY] LITERAL GET-TBL

9 [06 08 *] LITERAL J 2* + [' HV-C-TBL >BODY] LITERAL

10 GET-TBL HV

11 LOOP HV-OFF

12 LOOP ;

13 DECIMAL

14

15

109

0 \ TEST MODE SUPPORT

16FEB90CDB

1 HEX

2

3 : P-TST (S n -- flg)

4 0 DIG@ 73 AND = ;

5

6 : DO-TST? (S --)

7 [TST-BIT] LITERAL P-TST IF TEST-SEG THEN ;

8

9 : DO-TST-S? (S --)

10 [TSS-BIT] LITERAL P-TST IF TEST-SEG-S THEN ;

11

12 DECIMAL

13

14

15

110

0 \ TEST MODE SUPPORT

16FEB90CDB

1 HEX

2 ALSO ASSEMBLER

3 HERE ' CAL-TBL >BODY THERE !

4 ' NOOP , ' AD1-A-CAL , ' AD1-B-CAL , ' AD2-A-CAL ,

5 ' AD2-B-CAL , ' RTD-CAL , ' (LP) , ' TPM->BUF , ' (PG->BUF) , ' (SS->BUF) , ' DO-TST? , ' DO-TST-S? ,

6 ' (SS->BUF) , ' DO-TST? , ' DO-TST-S? , ' L-TM ,

7

8 DECIMAL

9

10 : (DO-EXEC) (S --)

11 NEW-CAL C@ NEW-CAL OFF DUP 1 12 BETWEEN

12 IF 2* [' CAL-TBL >BODY] LITERAL GET-TBL EXECUTE

13 ELSE DROP THEN ;

14

15

258

4MAY90CDB \ Pasp short test sequence

HEX

TEST-SEG-S (S --)

- Test sequence used to exercise all functions at least once.

Same sequence as TEST-SEQ except HV only samples 2 times

per step (vs 20 samples for TEST-SEQ).

4 * 98 + (6 * 107) + 446 = 1190 bytes

11 * 9 * (115) = 11,385

total = 12,575 bytes

DECIMAL

259

5JUL90CDB \ TEST MODE SUPPORT

P-TST (S n -- flg)

- Check for test connector code.

DO-TST? (S --)

- TEST-SEQ check.

DO-TST-S? (S --)

- TEST-SEQ-S check.

260

3JUL90CDB \ TEST MODE SUPPORT

HEX

CAL-TBL - Jump table for test routines.

' NOOP , ' AD1-A-CAL , ' AD1-B-CAL , ' AD2-A-CAL ,

' AD2-B-CAL , ' RTD-CAL , ' (LP) , ' TPM->BUF , ' (PG->BUF)

' (SS->BUF) , ' DO-TST? , ' DO-TST-S? ,

' (SS->BUF) , ' DO-TST? , ' DO-TST-S? , ' L-TM ,

(DO-EXEC) (S --)

(DO-EXEC) (S --)

- Execute routine pointed to by NEW-CAL.

Forth 83 Model

```

111
0 \ Pasp Command execution
16FEB90CDB
1 HEX
2
3 : USE-CMD (S cmd,flg -- )
4 IF DUP OF0 OR NEW-CMD C! 2* [ ' CMD-TBL >BODY ] LITERAL
5 GET-TBL EXECUTE ELSE DROP THEN ;
6
7 : GO-CMD (S -- )
8 NEW-CMD C@ DUP 0 F BETWEEN USE-CMD ;
9
10 : GO-STBY? (S -- )
11 NEW-CMD C@ DUP [ STBY-CMD ] LITERAL = USE-CMD ;
12
13 DECIMAL
14
15

```

```

112
0 \ Pasp Orbit definitions - Standby Mode
16FEB90CDB
1 DECIMAL
2
3 : STBY (S -- )
4 L-STBY [ INIT-SO ] LITERAL SP!
5 BEGIN
6 [ INIT-RO ] LITERAL RP! 4090 MILLISECONDS HK GO-CMD
7 FALSE UNTIL ;
8
9 : DO-EXEC (S -- ) (DO-EXEC) STBY ;
10
11 DECIMAL
12
13
14
15

```

```

113
0 \ Pasp Orbit definitions - Occultation
16FEB90CDB
1 HEX
2
3 : OCC (S -- )
4 L-OCC
5 HV-ON
6 30 0 DO
7 HK LP 14 [ -200V ] LITERAL [ OCC-CMD ] LITERAL HV
8 14 [ -000V ] LITERAL [ OCC-CMD ] LITERAL HV GO-STBY?
9 LOOP ;
10
11 DECIMAL
12
13
14
15

```

```

261
16FEB90CDB \ Pasp Command execution

```

```

USE-CMD (S cmd,flg -- )
- Use and execute the command pointer.

GO-CMD (S -- )
- If the command is valid use the command.

GO-STBY? (S -- )
- If the new command is standby do as commanded .

```

```

262
6JUL90CDB \ Pasp Orbit definitions - Standby Modes

```

```

STBY (S -- )
- Make a standby marker reset stack pointer .
Do endless loop of resetting the return pointer and sending
housekeeping data until a new command .
4 + k * 98 = total bytes ( k = # loops )

DO-EXEC (S -- )
- Execute routine pointed to by NEW-CAL.

```

```

263
15JUL90CDB \ Pasp Orbit definitions - Occultation

```

```

OCC (S -- )
- Occultation sequence. Forty-eight loops.

48 * ( 98 + 446 + 1087 + 1087 ) = 48 * 2718 = 130,464 bytes.

```

Forth 83 Model

114

0 \ Pasp Orbit definitions - Garrett Mode
16FEB90CDB

1 HEX

2

3 : GARRETT-MODE (S --)

4 L-GARR

5 HV-ON [INIT-S0] LITERAL SPI

6 BEGIN [INIT-R0] LITERAL RPI

7 HK LP 14 [-200V] LITERAL [OCC-CMD] LITERAL HV

8 14 [-000V] LITERAL [OCC-CMD] LITERAL HV GO-STBY?

9 FALSE UNTIL ;

10

11 DECIMAL

12

13

14

15

264

15JUL90CDB \ Pasp Orbit definitions - Garrett Mode

GARRETT-MODE (S --)

- Garrett sequence. Endless loop. Must exit with STBY command.

$$k * (98 + 446 + 1087 + 1087) = k * 2718 \text{ bytes. } k = \# \text{ loops}$$

115

0 \ Pasp Element Definition - 3 Per Orbit
16FEB90CDB

1 HEX

2

3 : (ORBIT-SEG) (S --)

4 4 0 DO

5 HK IV1 IV2 IV3 IV4 IV5 IV6 LP

6 HV-ON

7 OB 0 DO .RP 14 I 2* [' HV-TBL >BODY] LITERAL GET-TBL

8 HV-PTR @ 28 MIN 8 / 8 * J 2* + [' HV-C-TBL >BODY] LITERAL total = 14,775 bytes.

9 GET-TBL HV

10 I 6 = IF LP THEN LOOP

11 HV-OFF LOOP

12 HK IV1 IV2 IV3 IV4 IV5 IV6 LP ;

13

14 DECIMAL

15

265

4MAY90CDB \ Pasp Element Definition - 3 Per Orbit

: (ORBIT-SEG) (S --)

- For 4 elements

do hk, 6 ivs, lp

$$98 + 6 * 107 + 446 = 1186 \text{ bytes.}$$

$$11 * (1087) + 446 = 12,403 \text{ bytes.}$$

$$98 + 6 * 107 + 446 = 1186 \text{ bytes.}$$

116

0 \ Pasp Element Definition -
16FEB90CDB

1 DECIMAL

2

3 : ORBIT-SEQ (S n --)

4 DUP 1 6 BETWEEN NOT IF DROP 1 THEN DUP 1- 8 * HV-PTR !

5 L-ORBIT 1 L-SEG (ORBIT-SEG)

6 75 SECONDS 2 L-SEG (ORBIT-SEG)

7 75 SECONDS 3 L-SEG (ORBIT-SEG) OCC GO-CMD ;

8

9 DECIMAL

10

11

12

13

14

15

266

16FEB90CDB \ Pasp Element Definition -

ORBIT-SEQ (S n --)

- If n out of range then orbit 1 is executed.

$$4 + 3 * (4 + 14,775) + 130,464 = 174,805 \text{ bytes.}$$

Forth 83 Model

117

0 \ Pasp Orbit Definition -

16FEB90CDB

1 HEX

2 : ORBIT6 (S --) 6 ORBIT-SEQ STBY ;

3

bytes

4 : ORBIT5 (S --) 5 ORBIT-SEQ ORBIT6 ;

5

6 : ORBIT4 (S --) 4 ORBIT-SEQ ORBIT5 ;

7

8 : ORBIT3 (S --) 3 ORBIT-SEQ ORBIT4 ;

9

10 : ORBIT2 (S --) 2 ORBIT-SEQ ORBIT3 ;

11

12 : ORBIT1 (S --) 1 ORBIT-SEQ ORBIT2 ;

13

14 DECIMAL

15

118

0 \ PASP Command execution

16FEB90CDB

1 HEX

2

3 : BAD-CMD (S --)

4 NEW-CMD C@ [CMD-ERR1 100 *] LITERAL + ERROR! STBY ;

5

6 ALSO ASSEMBLER

7 HERE ' CMD-TBL >BODY THERE !

8 ' STBY , ' ORBIT1 , ' ORBIT2 , ' ORBIT3 ,

9 ' ORBIT4 , ' ORBIT5 , ' ORBIT6 , ' GARRETT-MODE ,

10 ' STBY , ' DO-EXEC , ' BAD-CMD , ' BAD-CMD , ' STBY , ' DO-EXEC , ' BAD-CMD , ' BAD-CMD ,

11 ' BAD-CMD , ' BAD-CMD , ' BAD-CMD , ' BAD-CMD , ' BAD-CMD , ' BAD-CMD , ' BAD-CMD , ' BAD-CMD ,

12

13 DECIMAL

14

15

119

0 \ PASP Interrupt Support -

16FEB90CDB

1 HEX

2 LABEL INT-HANDLER INT-CNT #) INC IRET

3 LABEL VEC-INIT

4 VEC-FIELD DO CX BX MOV 04 # BX SUB 08 # BX CMP 0=

5 IF NMI-INT # DX MOV

6 ELSE INT-HANDLER # DX MOV

7 THEN

8 DX 0 [BX] MOV CS 2 [BX] MOV 03 # CX SUB

9 LOOP RET

10

11 CODE SET-INT (S --)

12 BX PUSH VEC-INIT #) CALL BX POP NEXT C;

13 DECIMAL

14

15

267

16FEB90CDB \ Pasp Orbit Definition -

ORBITn (S --)

- Consecutive orbits will produce $6 * 174,805 = 1,048,830$

482 data bytes/ 512 telemetry bytes.

with overhead 1,114,110 bytes 8,912,879 bits

268

16FEB90CDB \ PASP Command execution

BAD-CMD (S --)

- Default command.

CMD-TBL - Command table.

' STBY , ' ORBIT1 , ' ORBIT2 , ' ORBIT3 ,
' ORBIT4 , ' ORBIT5 , ' ORBIT6 , ' GARRETT-MODE' STBY , ' DO-EXEC , ' BAD-CMD , ' BAD-CMD ,
' BAD-CMD , ' BAD-CMD , ' BAD-CMD , ' BAD-CMD ,

269

7JUL90CDB \ PASP Interrupt Support -

HEX

INT-HANDLER - Increment INT-CNT.

VEC-INIT

Registers Lost - BX,CX

- Initialize first 32 interrupt vectors. All point to
INT-HANDLER except NMI.

SET-INT (S --)

- Forth version of interrupt vector init.

Forth 83 Model

```

120
0 \ PASP Reset recovery Support -
16FEB90CDB
1 HEX
2
3 CODE (TEL-ON?) (S addr -- flg)
4 AX POP DI PUSH AX DI MOV
5 0 [DI] AL MOV AL 0 [DI] CMP 0<> IF 0 [DI] AL MOV THEN
6 1 [DI] AH MOV AH 1 [DI] CMP 0<> IF 1 [DI] AH MOV THEN
7 AX RDO-STAMP #) MOV
8 8E0E # AX CMP 0= IF TRUE # AX MOV ELSE FALSE # AX MOV THEN
9 DI POP 1PUSH C;
start
10
11 : TEL-ON? (S -- flg)
12 [ RDO-RAM ] LITERAL (TEL-ON?) ;
13
14 DECIMAL
15

```

```

121
0 \ PASP Reset recovery Support -
16FEB90CDB
1 HEX
2 : COLD-PWR? (S -- flg)
3 TEL-ON? NOT ;
4
5 : CLR-RAM (S -- )
6 [ DATA-START 2 + ] LITERAL [ 2700 DATA-START - ] LITERAL
7 ERASE ;
8
9 label (init-images)
10 dig #) ax mov ax dig-image #) mov
11 met #) ax mov ax met-image #) mov
12 rdo #) ax mov ax rdo-image #) mov ret
13 code init-images (init-images) #) call next c;
14 DECIMAL
15

```

```

122
0 \ PASP Reset recovery Support -
16FEB90CDB
1 HEX
2 : RES (S -- )
3 COLD-PWR?
4 IF
5 clr-ram NEW-CMD on T-BUF-INIT E-BUF-INIT init-images
6 1000 4.1-CNTR !
7 THEN
8 SET-INT
9 L-RESET STBY ;
10
11 : WDOG (S -- )
12 L-DOG RES
13 ( WDOG-FLG ON BEGIN GO-STBY? UNTIL ) ;
14 DECIMAL
15

```

```

270
18JUL90CDB \ PASP Reset recovery Support -

```

```

CLR-RAM (S -- )
- Initialize ram to 00.

```

```

CODE (TEL-ON?) (S addr -- flg)
Registers Lost - AX,DI

```

```

TEL-ON? (S -- flg)
- Used to detect initialized memory and determine of cold
  is required.

```

```

271
17JUL90CDB \ PASP Reset recovery Support -

```

```

HEX
COLD-PWR? (S -- flg)

```

```

WDOG (S -- )
- Make marker.
- Entry point if watch dog reset.

```

```

DECIMAL

```

```

272
17JUL90CDB \ PASP Reset recovery Support -

```

```

RES (S -- )
- Entry point if power on or manual reset.

```

Forth 83 Model

123

0 \ Initialization pasp system Low Level

16FEB90CDB

1 HEX

2

3

4 DECIMAL

5

6

7

8

9

10

11

12

13

14

15

124

0 \ Rom saver

16FEB90CDB

1 HEX

2 label t-nmi iret

3 LABEL COLD (COLD ENTRY)

with

4 CS AX MOV

5 AX SS MOV

6 INIT-S0 # SP MOV

7 AX DS MOV

8 AX ES MOV

9 T-NMI # 08 #) MOV

10 CS 0A #) MOV

11 INIT-R0 # RP MOV

12 BONE # CX MOV CL DIG 1 + #) MOV CH DIG 1 + #) MOV

13 00 # MET-CLR #) BYTE MOV

14

15

125

0 \ Rom saver

1 HEX

2 CMD 1 + #) AH MOV CMD 0 + #) AL MOV ax CMD-STAMP #) mov

3 EBEO # AX CMP 0<>

4 IF

5 ' WDOG >BODY # IP MOV

6 ELSE

7 00 # step #) mov 00 # CMD-CLK #) MOV

8 00 # LP-FLG #) MOV -1 # NEW-CMD #) MOV

9 00 # LP-CNTR #) MOV 00 # CMD-PTR #) MOV

10 (init-images) #) call

11 NMI-INT # 08 #) MOV (DIG-EN) #) CALL

12 ' RES >BODY # IP MOV

13 THEN NEXT

14

15

273

17JUL90CDB \ Initialization pasp system Low Level

COLD (COLD ENTRY)

- Entry point from 8086 reset vector. (FFF0)

Initialize cpu registers.

Feed watch dog.

Initialize stack, vector field and dig interface.

Clear met int and watch dog flag.

If NOT New reset command then WDOG.

ELSE RES. (pwr-on or reset command)

274

17JUL90CDB \ Rom saver

HEX

Fill field between last compiled byte and the reset vector

NOP's

install reset vector.

Fill rest with NOP's

HEX

275

18JUL90CDB

Forth 83 Model

126

276

```

0 \ Rom saver
1 HEX
2
3 HERE there 0FFFF HERE - 1+ 90 FILL
4 ALSO ASSEMBLER
5 DP-T @ 0FFFF DP-T !
6 COLD C-FIELD #) FAR JMP ( JUMP TO BOOT START CODE )
7 DP-T ! HEX
8
9
10
11
12
13
14
15

```

2JUL90CDB

127

277

```

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

```

15JUN90CDB

130

280

```

0 \ DACS BUS ADDRESSES - cont'
1 HEX
2
3 EXT-MEM-BASE 0400 + CONSTANT EXT-MEM-BASE
4 EXT-I/O-BASE 0B 4 * + CONSTANT TPI
5 EXT-I/O-BASE 0C 4 * + CONSTANT TPI-R
6
7 0 TPI + CONSTANT TPM-CMD 3 TPI + CONSTANT TPM-FMB
8 2 TPI + CONSTANT TPM-ADDR
9 0 TPI + CONSTANT TPM-CDR 2 TPI + CONSTANT TPM-DATA
10 3 TPI + CONSTANT TPM-STAT 5 TPI + CONSTANT TPM-RESET
11
12 DECIMAL
13
14
15

```

19JUN90CDB

Forth 83 Model

```

131
0 \ TPM INTERFACE BOARD CHECK OUT          19JUN90CDB 281
1 HEX
2 : milliseconds 0 do 10 0 do loop loop ;
3 : INIT-TPM (S -- )
4   OFF [ TPM-FMB ] LITERAL C! 19 MILLISECONDS
5   00 [ TPM-ADDR ] LITERAL C! [ TPM-DATA ] LITERAL C@
6   OFF XOR 01 AND 01 = NOT
7   IF OFF [ TPM-FMB ] LITERAL C! THEN ;
8
9 : READ-TPM (S -- ) base @ 4 base !
10  [ TPM-STAT ] LITERAL C@ OFF XOR 5 u.r cr
11  16 0 DO I [ TPM-ADDR ] LITERAL C! i 0b = if cr then
12  [ TPM-DATA ] LITERAL C@ OFF XOR 4 u.r LOOP base ! ;
13
14 DECIMAL
15

```

```

132
0 \ TPM INTERFACE BOARD CHECK OUT          19JUN90CDB 282
1 HEX
2
3 : ttst (S -- )
4   0b emit cr
5   OFF [ TPM-FMB ] LITERAL C! 1f4 MILLISECONDS READ-TPM
6   cr
7   OFF [ TPM-FMB ] LITERAL C! 1f4 MILLISECONDS READ-TPM ;
8
9 : tc tpm-cmd c! tpm-cmd c@ drop ;
10 DECIMAL
11
12
13
14
15

```

```

133
0 \ PASP SEQUENCE CONTROL BOARD - TERMINAL 19JUN90CDB 283
1 HEX
2
3 : T-TERMINAL (S -- ) \ EXIT ON ESC (1B hex)
4   dark BEGIN FALSE
5   P-KEY?
6   IF P-KEY 07F AND dup 0b =
7     if 0 12 at drop else EMIT then THEN
8   KEY?
9   IF KEY 07F AND
10    DUP 01B = IF TERM_EXIT \ EXIT LOOP
11    ELSE P-EMIT THEN THEN
12    UNTIL CR BEEP ." Leaving the Terminal Mode" ;
13 : ^[ t-terminal ;
14 DECIMAL
15

```

Forth 83 Model

147

297

```

0 \ HV SUPPORT
1 HEX
2 CODE DIG-EN
3 4 DO CX BX MOV OFF # 0DC08 [BX] BYTE MOV LOOP
4 OFF # 0DC07 #) BYTE MOV NEXT C;
5 : r-settle (s -- ) 800 0 do loop ;
6 : t-settle (s -- ) 4000 0 do loop ;
7 : DIG! (S data,addr -- ) 0dc08 + C! ;
8 : HV-OFF (S -- )
9 FF 2 DIG! R-SETTLE FF 1 DIG! R-SETTLE ;
10 : HV (S volts,cmd -- DUP 18 AND OFF XOR 1 DIG! R-SETTLE )
11 FF XOR 1 DIG! R-SETTLE 1F AND FF XOR 2 DIG! R-SETTLE ;
12 : hvt (s -- ) 1f 0c hv t-settle hv-off t-settle ;
13 : hvtt (s -- ) 00 0c hv t-settle 1f 0c hv t-settle ;
14 : tt (s n -- ) ['] hvtt 2dup 8 + ! 14 + ! ;
15 : oo hvtt hvt ;

```

5JUL90CDB

148

298

```

0 \
1 hex only forth also forth definitions
2 e02d constant unnest-t
3 e019 constant nest-t
4
5 code code-t (s cfa-t -- )
6 w pop 0 [w] jmp c;
7
8 code run-t (s cfa-t -- )
9 w pop nest-t #) jmp c;
10
11 decimal
12
13
14
15

```

5JUL90CDB

149

299

```

0 \
1 hex
2
3 : init-dacs fbae run-t ;
4
5 : cold fbfc code-t ;
6
7 : dig-en ee85 code-t ;
8
9 : dig@ ee99 run-t ;
10
11 decimal
12
13
14
15

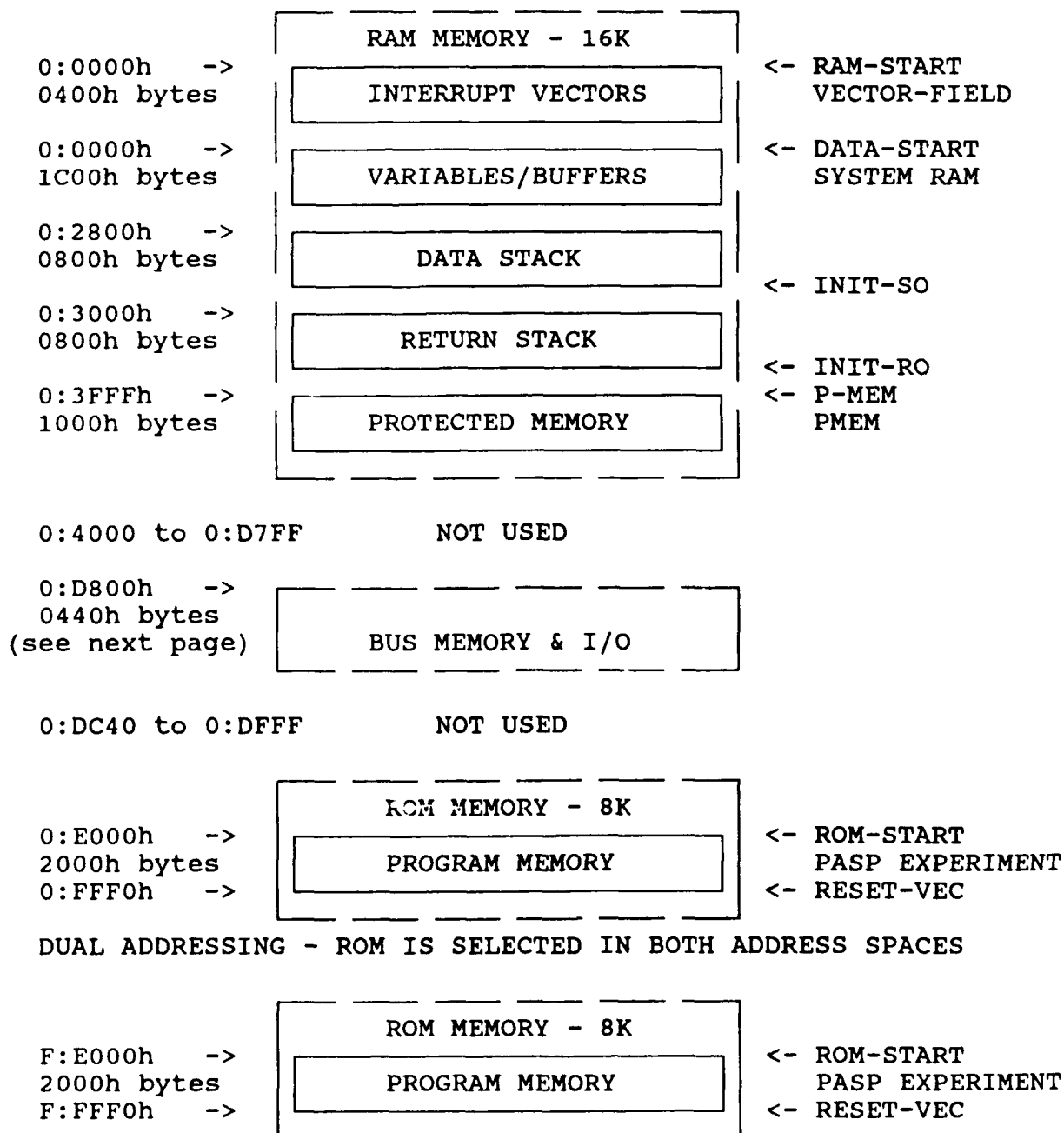
```

5JUL90CDB

Forth 83 Model

APPENDIX I - FLIGHT SOFTWARE MEMORY MAP

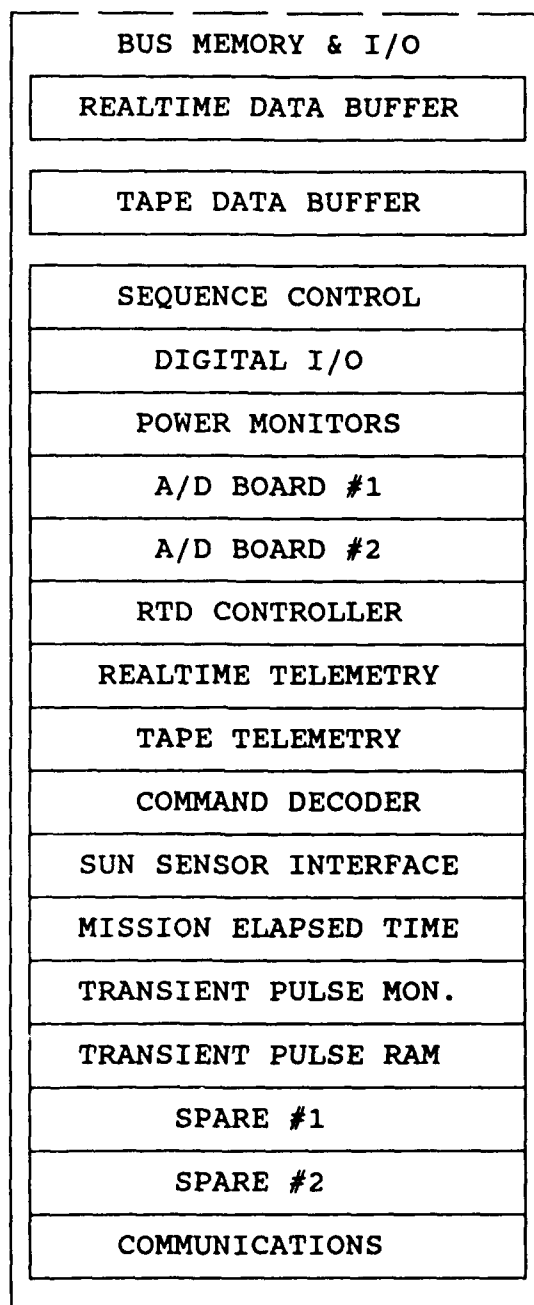
This memory map is prepared to aid in locating memory addresses and sizes. The comments in Appendix H - Flight Software Listing are also of interest. The first page of the memory map gives an overview. The second page contains details of the bus memory allocations. The final page is a screen dump of a metacompile session and shows used memory space for ROM memory (1C50) and RAM memory (100E).



0:D800h ->
0200h bytes

0:DA00h ->
0200h bytes

0:DC00h ->
0004h bytes
0:DC04h ->
0004h bytes
0:DC08h ->
0004h bytes
0:DC0Ch ->
0004h bytes
0:DC10h ->
0004h bytes
0:DC14h ->
0004h bytes
0:DC18h ->
0004h bytes
0:DC1Ch ->
0004h bytes
0:DC20h ->
0004h bytes
0:DC24h ->
0004h bytes
0:DC28h ->
0004h bytes
0:DC2Ch ->
0004h bytes
0:DC30h ->
0004h bytes
0:DC34h ->
0004h bytes
0:DC38h ->
0004h bytes
0:DC3Ch ->
0004h bytes



<- RDO-RAM
RDO-MEMORY

<- TDO-RAM
TDO-MEMORY

<- SEQ
SEQ-I/O

<- DIG
DIG-I/O

<- PM
PM-I/O

<- AD1
AD1-I/O

<- AD2
AD2-I/O

<- RTD
RTD-I/O

<- RDO
RDO-I/O

<- TDO
TDO-I/O

<- CMD
CMD-I/O

<- SSI
SSI-I/O

<- MET
MET-I/O

<- TPI
TPI-I/O

<- TPI-R
TPI-RAM

<- SPARE1
SPARE1-I/O

<- SPARE2
SPARE2-I/O

<- COMM
COMM-I/O

SCREEN PRINT OF METACOMPILE

SCR # 3

FA00 FA00 EE00 F200 EE00 B454 FC50 100E ED0A EDBE

3

Context: ASM86 ASM86 META FORTH ROOT

Current: ASM86

Unresolved references:

Statistics:

Last Host Address: B454

First Target Code Address: BC73

Last Target Code Address: D8C3

Target Size: 1C50

First Target RAM Address: 0

Last Target RAM Address: 100E

Context: META META FORTH ROOT

Current: META ok

APPENDIX J - WEIGHT OF PASP/Pegasus EXPERIMENT

1	Weight of PASP Plus	
1.1	Array Plate Assembly	
1.1.1	SP (silicon planar array)	1.02
1.1.2	GP (gallium arsenide planar)	1.59
1.1.3	CC (Cassegrainian concentrator)	2.0
1.1.4	SC (SLATS concentrator)	4.24
1.1.5	BC (Boeing light funnel?) (est.)	2 . 0
1.1.6	Planar array (TBD) (est.)	1 . 2
1.1.7	Langmuir probe head	0.47
1.1.8	Transient pulse monitor antenna	0.25
1.1.9	Sun sensor head	0.56
1.1.10	Pressure gauge head	7.60
1.1.11	Cables and connectors	2.03
1.1.12	Subtotal	22.96
1.2	Box Assembly (or radiation shielding)	
1.2.1	Marshall half-height EAC	24.4
1.2.2	Additional cold plates and structure	10.0
1.2.3	Thermal shielding and MLI	10.0
1.2.4	Subtotal	34.4
1.3	DACS	
1.3.1	DACS chassis	14.81
1.3.2	DACS boards	4.45
1.3.3	Cabling	3.67
1.3.4	SPS and HK boards	2.35
1.3.5	Subtotal	25.28
1.4	ASIS	
1.4.1	ASIS chassis	11.24
1.4.2	ASIS boards	8.85
1.4.3	Cabling	3.70
1.4.4	APS and HK boards	2.35
1.4.5	Subtotal	26.14
1.5	GFE Electronics and Accessories	
1.5.1	Pressure gauge electronics	2.10
1.5.2	Langmuir probe electronics	3.762
1.5.3	Sun sensor electronics	0.87
1.5.4	Transient pulse monitor	12.65
1.5.5	Pressure sensor	1.01
1.5.6	Emitter	1.03
1.5.7	Subtotal	21.42
1.6	Total weight	140.2 lbs.

APPENDIX K - GSE MENU SOFTWARE LISTING

Software for operation of the GSE was completed late in the PASP Plus program. One reason for the delay was the need to incorporate a more user-friendly menu system so that the GSE could be easily operated by someone other than the software development engineer. The menu system was complex enough that a new version of FORTH, the F-PC (November 1989 release), was obtained and used.

The menu software listing does not have shadow screens to assist in following the code. This should not be a serious impediment to someone familiar with FORTH since a number of comments are included.

Other software is also required to operate the GSE and perform activities such as metacompile and PROM burning. This software includes:

1. F83 version of FORTH (modified slightly for PASP);
2. F-PC (November 1989 release) version of FORTH; and
3. Metap.blk.

The PASP internal command structure was modified when the TPM instrument was added. The TPM required too many commands for the 16 available command words in the IMPS ICD requirements. This problem was overcome by multiplexing. The tpmm-cmd (constant 0C in screen 26 of the flight software) alerts PASP that the next command is for the TPI interface board and is a TPM command. In a similar fashion two other commands can now be used for multiplexing, the csel-cmd and the psel-cmd (see constants 0A and 0B in screen 26). Csel allows up to 16 commands to be selected while psel allows up to 16 parameters to be selected. The exec-cmd (constant 09 in screen 26) is used to execute either of the two multiplex commands. As a consequence of this change it is now possible to perform in-flight calibrations and other system test sequences and instrument operations (See bottom of page 9 of this GSE Menu listing for commands). Examples of parameters that can be selected are IV channels, sun sensor angles for test, or cal(ibration) channels.

\ GSEMENU.SEQ Visual menu selection tool for PASP GSE by C.D. Boswell

only forth also definitions

empty

include addon.seq

only forth also definitions

vocabulary p_gse

editor also p_gse definitions

postfix

headerless

DECIMAL

```
: (sys) ( addr -- )
  [ also hidden ] $sys 0 24 at cr ?syserror ;
```

```
: new-dev-rom ( -- )
  " cd\gse" ">$ (sys)
  " c:\gse\mkpasp" ">$ (sys)
  " cd\fpc" ">$ (sys) ;
```

HEX

```
0 value i-buff
0 value o-buff-l
0 value o-buff-h
```

```
: ">0$ ( addr,len -- addr')
  over + 1- 0 swap c! ;
```

```
: ">d$ ( addr,len -- addr')
  over + 1- 0d swap c! ;
```

```
: read-file ( seg,addr,n,faddr -- )
  seqhandle+ $>handle
  seqhandle+ HOPEN ABORT" Open ERR!"
  ROT
  seqhandle+ SWAP EXHREAD 0= ABORT" Read ERR!"
  seqhandle+ HCLOSE ABORT" Close ERR!" ;
```

only forth also editor also p_gse definitions

\$8000 value rom-size

```

: make-split ( -- )
  i-buff rom-size u2/ 0
  do dup
    i 2* @L split
    o-buff-h i C!L
    o-buff-l i C!L
  loop drop ;

: write-pasp ( n, faddr -- )
  seqhandle+ $>handle
  seqhandle+ HDELETE          DROP
  seqhandle+ HCREATE          ABORT" Write-Pasp Create ERR!"
  dup 4 =
  if drop
    0000 rom-size i-buff
  else
    2 /mod
    if rom-size u2/ u2/ else 0000 then rom-size u2/ u2/ rot
    if o-buff-h else o-buff-l then
    then
    seqhandle+ swap EXHWRITE 0= ABORT" Write-Pasp Write ERR!"
    seqhandle+ HCLOSE          ABORT" Write-Pasp Close ERR!" ;

```

```

variable ss-temp
variable sp-temp
0000 constant hex-entry
0 value h-buff

```

```

code fcall-ext ( seg,addr,seg -- )
  ax pop
  ax push
  bx push
  cx push
  dx push

  bp push
  si push
  di push

  ds push
  es push
  ax bp mov

  sp bx mov
  sp sp-temp #) mov
  ss ss-temp #) mov

  far [] 12 [bx] call

```

```

cs:
ss-temp #) ss mov
cs:
sp-temp #) sp mov

```

```

es pop
ds pop

```

```

di pop
si pop
bp pop

```

```

dx pop
cx pop
bx pop
ax pop
ax pop
ax pop
next c;

```

only forth also editor also p_gse also hidden also

```

: make-hex ( addr,len -- )
  " OBJ2HEX.OBJ " ">0$ findfirst 0 = not
    if true abort" obj2hex.obj is missing" then
  2dup ">0$ findfirst 0 =
  if
    &> h-buff 0B00 alloc-mem
    h-buff 0000 0B00
    " OBJ2HEX.OBJ" ">$ read-file \ MUST BE PRESENT ON CURRENT DIR
    2dup ">d$ drop
    ?cs: -rot h-buff 87 + 05b rot cmovel
    h-buff 10 + hex-entry over fcall-ext
    &> h-buff dealloc-mem
  else
    true abort" source file not found "
  then ;

```

```

: MAKE-DEV ( -- )
  8000 %!> rom-size
  " F83-D1.COM " ">0$ findfirst 0 = not
    if true abort" f83-d1.com is missing" then
  " F83-D2.COM " ">0$ findfirst 0 = not
    if true abort" f83-d2.com is missing" then
  " PASPRAM.OBJ " ">0$ findfirst 0 = not
    if true abort" paspram.obj is missing" then

  &> i-buff rom-size alloc-mem
  i-buff 0000 0100
  " F83-D1.COM" ">$ read-file \ MUST BE PRESENT ON CURRENT DIR

```

```

i-buff 0100 7F00
" F83-D2.COM" ">$ read-file \ MUST BE PRESENT ON CURRENT DIR
i-buff 6000 10 - 2000
" PASPRAM.OBJ" ">$ read-file \ MUST BE PRESENT ON CURRENT DIR

4 " DEVROM00.OBJ" ">$ write-pasp
  &> o-buff-l rom-size u2/ alloc-mem
  &> o-buff-h rom-size u2/ alloc-mem
  make-split
0 " DEVROM1L.OBJ" ">$ write-pasp
1 " DEVROM1H.OBJ" ">$ write-pasp
2 " DEVROM2L.OBJ" ">$ write-pasp
3 " DEVROM2H.OBJ" ">$ write-pasp
  &> o-buff-h dealloc-mem
  &> o-buff-l dealloc-mem
  &> i-buff dealloc-mem
  " DEVROM1L.OBJ " make-hex
  " DEVROM1H.OBJ " make-hex
  " DEVROM2L.OBJ " make-hex
  " DEVROM2H.OBJ " make-hex ;

: MAKE-PASP ( -- )
  2000 %!> rom-size
  " PASPROM.OBJ " ">0$ findfirst 0 = not
  if true abort" pasprom.obj is missing" then

  &> i-buff rom-size alloc-mem
  i-buff 0000 rom-size
  " PASPROM.OBJ" ">$ read-file \ MUST BE PRESENT ON CURRENT DIR
4 " SEQROMS.OBJ" ">$ write-pasp
  &> o-buff-l rom-size u2/ alloc-mem
  &> o-buff-h rom-size u2/ alloc-mem
  make-split
0 " SEQROM1L.OBJ" ">$ write-pasp
1 " SEQROM1H.OBJ" ">$ write-pasp
2 " SEQROM2L.OBJ" ">$ write-pasp
3 " SEQROM2H.OBJ" ">$ write-pasp
  &> o-buff-h dealloc-mem
  &> o-buff-l dealloc-mem
  &> i-buff dealloc-mem
  " SEQROM1L.OBJ " make-hex
  " SEQROM1H.OBJ " make-hex
  " SEQROM2L.OBJ " make-hex
  " SEQROM2H.OBJ " make-hex ;

```

comment:

```

: aonly ( --- )
  savescr cursor-off
  ['] noop save!> dobutton

```

```

14 6 64 9 box&fill
." you must use the \1 alt \0 key sequence for macros. "
bcr
." \s10press a \r key \0 to continue."
key drop
restore> dobutton
restscr cursor-on ;

```

```
comment;
```

```

0 value exit-flg
0 value return-flg

```

```

: menu-exit ( -- )
  %on> return-flg
  %on> exit-flg ;

```

```

: return ( -- )
  %on> return-flg ;

```

```
hex
```

```
\ ibm imps simulator addresses
```

```

$3e0 constant gse
gse 0 + constant gse-stat
gse 1 + constant cmd-ls
gse 2 + constant cmd-ms
gse 3 + constant met-ls
gse 4 + constant met-ms
gse 5 + constant met-sync
gse 6 + constant tape-int
gse 7 + constant gse-bus

```

```

gse 0 + constant ssi-ata
gse 1 + constant ssi-x
gse 2 + constant ssi-y

```

```

1 constant tape-bit
2 constant dma-bit
4 constant g-data-bit
variable stat-image

```

```
\ real time display
```

```

: ssy-fix ( n -- n' ) \ Compensate for GSE swap nibbles in Y display
  $10 /mod swap $10 * + ;

```

```
code ss-fix ( n -- n' ) \ Compensate for bits swapped in the dacs ssi board
```



```

ax pop
al ah mov
$04 # al test 0= if $f7 # ah and else $08 # ah or then
$08 # al test 0= if $fb # ah and else $04 # ah or then
$40 # al test 0= if $7f # ah and else $80 # ah or then
$80 # al test 0= if $bf # ah and else $40 # ah or then
ah al mov
00 # ah mov
lpush c;

```

only forth also editor also p_gse also hidden also

```

: sso-fix ( n -- n' ) \ Compensate for GSE swap nibbles in Y display
  split ss-fix
  swap ss-fix ssy-fix swap join ;

```

```

: sso@ ( -- x,y,ata)
  ssi-x pc@ ss-fix
  ssi-y pc@ ss-fix ssy-fix
  ssi-ata pc@ 01 and ;

```

```

: dacs-meta      noop ;
: cal-files      noop ;
: real-setup     noop ;

```

```

: iv-cal         noop ;
: etelem-cal     noop ;
: itelem-cal     noop ;
: elect-cal      noop ;
: emitter-cal    noop ;
: ad1a-tbl       noop ;
: ad1b-tbl       noop ;
: ad2a-tbl       noop ;
: ad2b-tbl       noop ;
: rtd-tbl        noop ;
: dacs-ptbl      noop ;
: asis-ptbl      noop ;
: ss-cal         noop ;
: lp-cal         noop ;
: pg-cal         noop ;
: tpm-cal        noop ;

```

```

$00 constant spr1-cmd
$01 constant orb1-cmd
$02 constant orb2-cmd
$03 constant orb3-cmd
$04 constant orb4-cmd
$05 constant orb5-cmd

```

```

$06 constant orb6-cmd
$07 constant garr-cmd
$08 constant stby-cmd
$09 constant exec-cmd
$0a constant csel-cmd
$0b constant psel-cmd
$0c constant tpmm-cmd
$0d constant lpwr-cmd
$0e constant mres-cmd
$0f constant spr2-cmd

```

```

label .2sec-wait ( -- )
4 # cx mov here
  cx push
  $2000 # cx mov
  here loop
  cx pop
loop ret

```

only forth also editor also p_gse also hidden also

```

: code->w ( code -- w )
  111 * cf xor ;

```

```

code (cmds) ( n -- )
ax pop
cmd-ls # dx mov
al dx out
dx inc
ah al mov
al dx out
.2sec-wait #) call
next c;

```

```

: cmds ( code -- )
  $0f and code->w (cmds) ;

```

```

: cmd-t ( cmd -- )
  tpmm-cmd cmds cmds ;

```

```

: cmd-p ( psel -- )
  psel-cmd cmds cmds ;

```

```

: cmd-c ( csel -- )
  csel-cmd cmds cmds exec-cmd cmds ;

```

```

code met! ( n -- )
ax pop
ax ror
met-ls # dx mov
al dx out

```

```

dx inc
ah al mov
al dx out
.2sec-wait #) call
next c;

```

```

code (set-met) ( -- w1,w2,w3)

```

```

$2c # ah mov
$21 int
$1000 # ax mov
ch al mov
$40 # al or
ax push
cl ch mov
dh shl
dh cl mov
cx shr
cx shr
cx shr
$20 # ch or
cx push
0a # al mov
dl mul
dh shl
dh ah or
0f # ah and
$30 # ah or
lpush c;

```

```

: set-met ( -- )
(set-met) rot met! swap met! met! 0 met-sync pc! ;

```

only forth also editor also p_gse also hidden also

decimal

2variable box-corner

0 value pos<-dacs-cmds

```

: set<-dacs-cmds ( n -- )
%!> pos<-dacs-cmds
3 12 bounds
do
box-corner 2@ 23 i d+ at
pos<-dacs-cmds i = if " <- " else " " then type
loop ;

```

```

: dacs-cmds ( --- )
  savescr
  [''] noop save!> dobutton
  ?doingmac 0= \ if we are doing a macro, don't display
                \ command menu box.
  if screenline 1+ dup 12 >
    if 13 -
    then 18 swap 58 over 17 +
          2over box-corner 2!
          box&fill

          ."          dacs commands menu          key " bcr bcr
          ." Standby command          S " bcr
          ." orbit1 command          1 " bcr
          ." orbit2 command          2 " bcr
          ." orbit3 command          3 " bcr
          ." orbit4 command          4 " bcr
          ." orbit5 command          5 " bcr
          ." orbit6 command          6 " bcr
          ." Garrett command          G " bcr
          ." Dacs power on          D " bcr
          ." Warm restart          W " bcr
          ." Low power mode          L " bcr
          ." Initialize met          I " bcr
          ." Quit          Q " bcr
          ." \s10\r esc \0 = cancel"
  pos<-dacs-cmds set<-dacs-cmds
then
begin
  key bl or >r
  's' r@ = if 08 cmds 3 set<-dacs-cmds then
  r@ dup '1' '6' between
  if dup cmds '0' - 3 + set<-dacs-cmds
  else drop then
  'g' r@ = if 07 cmds 10 set<-dacs-cmds then
  'd' r@ = if 14 CMDS
          14 cmds 11 set<-dacs-cmds then
  'w' r@ = if 14 cmds 12 set<-dacs-cmds then
  'l' r@ = if 13 cmds 13 set<-dacs-cmds then
  'i' r@ = if set-met 14 set<-dacs-cmds then
  59 r@ = if menu-exit r>drop 113 >r then
  r> 113 =
  until
  restscr
  restore> dobutton
  sdisplay showstat cursor-on ;

```

```

$1 value ad1-a-cal
$2 value ad1-b-cal
$3 value ad2-a-cal
$4 value ad2-b-cal

```

```

$5 value rtd-cal
$6 value (lp)
$7 value tpm->buf
$8 value (pg->buf)
$9 value (ss->buf)
$a value do-tst?
$b value do-tst-s?
$c value l-tm

```

comment:

```

: (do-exec) (s -- )
  new-cal c@ new-cal off dup 1 11 between
  if 2* [ ' cal-tbl >body ] literal get-tbl execute
  else drop then ;

```

comment;

```

: test-seg      do-tst?      cmd-c ;
: test-seg-s    do-tst-s?    cmd-c ;
: lp-data       (lp)         cmd-c ;
: pg-data       (pg->buf)     cmd-c ;
: ss-data       (ss->buf)     cmd-c ;
: tpm-data      tpm->buf      cmd-c ;
: time-mkr      l-tm         cmd-c ;

```

0 value pos<-test-cmds

```

: set<-test-cmds ( n -- )
  %!> pos<-test-cmds
  3 7 bounds
  do
    box-corner 2@ 28 i d+ at
    pos<-test-cmds i = if " <- " else " " then type
  loop ;

```

: test-cmds

```

  savescr
  ['] noop save!> dobutton
  ?doingmac 0= \ if we are doing a macro, don't display
               \ command menu box.
  if screenline 1+ dup 12 >
    if 13 -
    then 18 swap 58 over 12 +
         2over box-corner 2!
         box&fill

  ."          test commands menu          key " bcr bcr
  ." Normal test sequence                 N " bcr
  ." Abbreviated test sequence            A " bcr
  ." Langmuir probe data                  L " bcr

```

```

        ."      Pressure gauge data          P " bcr
        ."      Sun sensor data             S " bcr
        ."      Tpm data                    T " bcr
        ."      time Marker                 M " bcr
        ."      Quit                        Q " bcr
        ."      \s10\r esc \0 = cancel"
        pos<-test-cmds  set<-test-cmds
    then
    begin
        key bl or >r
        'n' r@ = if test-seg 3 set<-test-cmds then
        'a' r@ = if test-seg-s 4 set<-test-cmds then
        'l' r@ = if lp-data 5 set<-test-cmds then
        'p' r@ = if pg-data 6 set<-test-cmds then
        's' r@ = if ss-data 7 set<-test-cmds then
        't' r@ = if tpm-data 8 set<-test-cmds then
        'm' r@ = if time-mkr 9 set<-test-cmds then
        59 r@ = if menu-exit r> drop 113 >r then
        r> 113 =
    until
    restscr
    restore> dobutton
    sdisplay showstat cursor-on ;

```

```

ascii 0 value tpm-cmd
ascii 0 value tpm-cmd-s

```

```

: .tpm-cmd ( -- )
    box-corner 2@ 22 3 d+ at tpm-cmd emit
    box-corner 2@ 22 4 d+ at tpm-cmd-s emit ;

```

```

: tpm-select ( n -- )
    %!> tpm-cmd .tpm-cmd ;

```

```

: send-tpm-cmd ( -- )
    tpm-cmd dup cmd-t %!> tpm-cmd-s .tpm-cmd ;

```

```

: tpm-cmds

```

```

    savescr
    ['] noop save!> dobutton
    ?doingmac 0= \ if we are doing a macro, don't display
                  \ command menu box.
    if      screenline 1+ dup 12 >
        if      13 -
        then    18 swap 58 over 7 +
                2over box-corner 2!
                box&fill

```

```

        ."      tpm commands menu          key " bcr bcr

```

```

        ."      select tpm command          0-7 " bcr
        ."      Send tpm command           S " bcr
        ."      Quit                       Q " bcr
        ."      \s10\r esc \0 = cancel"
        .tpm-cmd

```

```

then
begin
  key bl or >r
  r@ dup '0' '7' between
  if tpm-select else drop then
  's' r@ = if send-tpm-cmd then
  59 r@ = if menu-exit r>drop 113 >r then
  r> 113 =
until
restscr
restore> dobutton
sdisplay showstat cursor-on ;

```

```

: ad1a-pt      ad1-a-cal cmd-c ;
: ad1b-pt      ad1-b-cal cmd-c ;
: ad2a-pt      ad2-a-cal cmd-c ;
: ad2b-pt      ad2-b-cal cmd-c ;
: rtd-pt       rtd-cal  cmd-c ;

```

```

      0 value pos<-cal-cmds
ascii 0 value cal-chan

```

```

: set<-cal-cmds ( n -- )
  %!> pos<-cal-cmds
  4 5 bounds
  do
    box-corner 2@ 25 i d+ at
    pos<-cal-cmds i = if " <- " else " " then type
  loop ;

```

```

: .cal-chan ( -- )
  box-corner 2@ 22 3 d+ at cal-chan emit
  pos<-cal-cmds set<-cal-cmds ;

```

```

: cal-select ( n -- )
  dup cmd-p %!> cal-chan .cal-chan ;

```

```

: cal-cmds
  savescr
  ['] noop save!> dobutton
  ?doingmac 0= \ if we are doing a macro, don't display
                \ command menu box.
  if      screenline 1+ dup 12 >
    if      13 -

```

```

then      18 swap 58 over 11 +
          2over box-corner 2!
          box&fill

```

```

."          cal  commands menu          key " bcr bcr
."      select cal channel              0-7 " bcr
."      ad1 mux a data packet          alt-f1 " bcr
."      ad1 mux b data packet          alt-f2 " bcr
."      ad2 mux a data packet          alt-f3 " bcr
."      ad2 mux b data packet          alt-f4 " bcr
."      Rtd data packet                R " bcr
."      Quit                          Q " bcr
."      \s10\r esc \0 = cancel"
.cal-chan

```

```

then
begin
  key bl or >r
  r@ dup '0' '7' between
    if cal-select
    else drop
  $e8 r@ = if ad1a-pt 4 set<-cal-cmds then
  $e9 r@ = if ad1b-pt 5 set<-cal-cmds then
  $ea r@ = if ad2a-pt 6 set<-cal-cmds then
  $eb r@ = if ad2b-pt 7 set<-cal-cmds then
  'r' r@ = if rtd-pt 8 set<-cal-cmds then
  59 r@ = if menu-exit r>drop 113 >r then
  r> 113 =
until
restscr
restore> dobutton
sdisplay showstat cursor-on ;

```

```
'1' value iv-cmd
```

```
0 value pos<-asis-cal
```

```

: set<-asis-cal ( n -- )
  %!> pos<-asis-cal
  4 5 bounds
  do
    box-corner 2@ 26 i d+ at
    pos<-asis-cal i = if " <- " else " " then type
  loop ;

: .iv-cmd ( -- )
  box-corner 2@ 22 3 d+ at iv-cmd emit
  box-corner 2@ 10 4 d+ at iv-cmd emit
  pos<-asis-cal set<-asis-cal ;

```



```
: iv-select ( n -- )
  %!> iv-cmd .iv-cmd ;
```

```
: asis-cal
```

```
  savescr
  ['] noop save!> dobutton
  ?doingmac 0= \ if we are doing a macro, don't display
               \ command menu box.
  if      screenline 1+ dup 12 >
    if      13 -
    then    18 swap 58 over 12 +
            2over box-corner 2!
            box&fill
```

."	asis cal menu	key "	bcr bcr
."	select iv array #	1-6 "	bcr
."	make iv cal table	V "	bcr
."	make E telem cal table	E "	bcr
."	make I telem cal table	I "	bcr
."	make elect cal table	L "	bcr
."	make eMitter cal table	M "	bcr
."	cal File menu	F "	bcr
."	Quit	Q "	bcr
."	\s10\r esc \0 = cancel"		
.	iv-cmd		

```
  then
  begin
    key bl or >r
    r@ dup '1' '6' between
      if iv-select
      else drop then
        'v' r@ = if iv-cmd drop iv-cal 4 set<-asis-cal then
        'e' r@ = if etelem-cal 5 set<-asis-cal then
        'i' r@ = if itelem-cal 6 set<-asis-cal then
        'l' r@ = if elect-cal 7 set<-asis-cal then
        'm' r@ = if emitter-cal 8 set<-asis-cal then
        'f' r@ = if cal-files then
        59 r@ = if menu-exit r>drop 113 >r then
        r> 113 =
      until
    restscr
    restore> dobutton
    sdisplay showstat cursor-on ;
```

```
: make-file ( h-addr>,faddr -- dhand )
  seqhandle+ $>handle
  seqhandle+ hdelete drop
  seqhandle+ hcreate abort" create file err!"
  seqhandle+ swap !
```

```

seqhandle+ >hndle @ ;

: open-file ( h-addr>,faddr-- dhnd )
  seqhandle+ $>handle
  seqhandle+ hopen          abort" open file err!"
  seqhandle+ swap !
  seqhandle+ >hndle @ ;

0 value #dbuf-rec

: write-file ( seg,addr,n,h-addr> -- )
  >r rot r> swap over
  if
    exhwrite dup %!> #dbuf-rec
    0= abort" write file err!"
  else
    true abort" file not open"
  then ;

\ include r488.seq

0 value source-flg
-1 value save-flg
-1 value saveb-flg
0 value view-flg
0 value tele-mode
3 value tele-max

hex

code gc! ( b -- )
ax pop
$ff # al xor
gse-bus # dx mov
al dx out
next c;

code gc@ ( -- b)
1 # cx mov here loop
gse-bus # dx mov
dx al in
0 # ah mov
$ff # al xor
ax push
next c;

only forth also editor also p_gse also hidden also

```

```

: stat! ( b -- )
  dup stat-image c! gse-stat pc! ;

: g-adr ( -- )
  stat-image c@ g-data-bit ff xor and stat! ;

: g-dat ( -- )
  stat-image c@ g-data-bit or stat! ;

: g-chk ( addr,offset -- )
  g-adr + gc! g-dat gse-stat pc@ 04 and
  ( 0 = if true abort" gse select error " then ; )
  dup 0 = if cr " gse select error " h. else drop then ;

: dc! ( b,addr -- ) $11 g-chk gc! ;

: dc@ ( addr -- b) $13 g-chk 0 gc! gc@ ;

```

```

2variable old-vec1
2variable old-vec2

```

```

code get-int (s int# - cs,addr ) \ stack int# addr
  ax pop

```

```

  ds push
  es push
  bp push
  si push

```

```

  $35 # ah mov
  $21 int \ use int service
  es dx mov
  bx ax mov

```

```

  si pop
  bp pop
  es pop
  ds pop
  2push c;

```

```

code set-intds (s cs:,addr,int# - ) \ set int#
  ax pop
  dx pop
  ds pop
  bx push
  $25 # ah mov
  $21 int
  bx pop

```

```

cs push
ds pop
next c;

```

```

code set-tnr (s mem-size u16/ -- ) \ set int#
dx pop
$31 # ah mov
$21 int c;

```

```

\ ibm interface board support - error buffer support 5jun90cdb
hex
$20 constant e-buf-size

```

```

variable e-buf-start e-buf-size 2- allot
2variable e-buf-wrt
2variable e-buf-rd
variable e-buf-gauge

```

```

label init-e-buf (s -- )
00 # e-buf-wrt #) mov
00 # e-buf-wrt 2 + #) mov
00 # e-buf-rd #) mov
00 # e-buf-rd 2 + #) mov
00 # e-buf-gauge #) mov
e-buf-size # cx mov here
cx di mov
di dec
di dec
$544d # e-buf-start [di] word mov
cx dec
loop
ret

```

```

label sys-err (s -- )
e-buf-wrt #) di mov
ax e-buf-start [di] mov
di inc
di inc
e-buf-size # di cmp <=
if
00 # di mov
e-buf-wrt 2+ #) inc
then
di e-buf-wrt #) mov
e-buf-gauge #) inc
ret

```

only forth also editor also p_gse also hidden also

```
code error! (s err# -- )
  ax pop
  sys-err #) call
  next c;
```

```
label rd-e-buf (s -- )
  e-buf-gauge #) 00 # cmp >
  if
    e-buf-gauge #) dec
    e-buf-rd #) di mov
    e-buf-start [di] ax mov
    di inc
    di inc
    e-buf-size # di cmp <=
    if
      00 # di mov
      e-buf-rd 2+ #) inc
    then
      di e-buf-rd #) mov
    else
      false # ax mov
    then
      ret
```

only forth also editor also p_gse also hidden also

```
code error? (s -- code/false )
  rd-e-buf #) call
  ax push
  next c;
```

```
label dos-call (s -- )
  $21 int u<
  cs ax mov
  ax ds mov
  if
    ax neg sys-err #) call
  then
    ret
```

\ ibm interface board support - rdo buffer support 7sep89cdb
hex

only forth also p_gse definitions

```
$100 $10 * constant rdo-size
0 value rdo-ram
```

```
variable rdo-wrt
variable rdo-rd
variable rdo-ptr
```

```
variable rdo-gauge
variable rdo-handle
variable rdo-cnt
```

\ increment if reading from cap file

```
: init-rdo (s -- )
  &> rdo-ram rdo-size alloc-mem
  rdo-ram 0
  $1000 um/mod drop rdo-size u16/ + 0
  $1000 um/mod 0 <> \ check for dma memory page boundary
  if
    rdo-size 2 * swap $10 * - dup
    rdo-ram swap u16/ setblock 0 <> \ if there is an overlap then add
memory
    if
      true abort" dma alloc error "
    else \ if successful move buffer pointer
      rdo-size - u16/ %!> rdo-ram
    then
  else drop
    rdo-wrt off
    rdo-rd off
    rdo-ptr off
    rdo-gauge off
    rdo-cnt off
    rdo-ram 0 rdo-size bounds
    do $ff over i c!l loop drop
  then ;

code rdo-seg ( -- addr )
  &> rdo-ram #) 0 # cmp 0=
  if
    >h
    beep ." rdo not allocated"
    init-rdo
    h>
  then
  &> rdo-ram #) push
  next c;

: rdo-c! ( n,addr -- )
  rdo-seg swap c!l ;
```

```

: rdo-! ( n,addr -- )
  rdo-seg swap !l ;

: rdo-c@ ( addr -- n )
  rdo-seg swap c@l ;

variable rdo-flg
variable flush-rdo?
0 value rdo-h-addr
0 value #rdo-rec

: make-rdo (s -- / create rdo-00.cap)
  flush-rdo? off
  &> rdo-h-addr " rdo-00.cap " ">$ make-file rdo-handle !
  rdo-flg on ;

: open-rdo (s -- / open rdo-00.cap)
  flush-rdo? off
  &> rdo-h-addr " rdo-00.cap " ">$ open-file rdo-handle !
  rdo-h-addr endfile $80 * swap u16/ u16/ u2/ + %!> #rdo-rec
  0. rdo-h-addr movepointer
  rdo-flg on ;

code close-rdo
  false # flush-rdo? #) mov
  false # rdo-flg #) mov
  $3e00 # ax mov \ close function call
  rdo-handle #) bx mov \ get file handles-call
  dos-call #) call
  next c;

: dump-rdo ( start -- )
  [ also editor ]
  depth 1 < if 1 then 1- \ default to rec 1
  init-rdo
  open-rdo
  rdo-h-addr edi hnd1 $>handle \ copy file to edit handle
  dup #rdo-rec 1+ u> not
  if
    ?printer.ready 0=
    if
      .noprinter
      exit
    then
    printer-init
    setfile date&time
    dup 0 $200. d* rdo-h-addr movepointer
    #rdo-rec
    dup lastpage ! 1+
    swap
    dup 1+ pagenumber !
    base @ >r hex

```

```

2dup u>
if
?do
  0 $200 rdo-h-addr rdo-seg exhread
  $200 = not
  if
    beep
    ." rdo-dump read error"
    leave
  then
  key?
  if
    key $bb =          \ exit if f1
    if
      leave
    then
  then
  rdo-ram 0 $200

  printing on
  .underline
  $a 0 do cr loop ldump $a 0 do cr loop
  decimal .footer hex
  newpage
  printing off

  loop
then
r> base !
then
printing off
close-rdo ;

: ty    dump-rdo ;

\ ibm interface board support - dma support          3sep89cdb
hex

$0200 constant dma-count
true constant rdo-input

variable 2sec-cnt
variable do-poll-func
variable dma-base
variable cap-cnt          \ increment if reading from cap file

label set-cap
0 [di] dx mov          \ get rdo buffer address
dma-count # cx mov     \ get count
$3f00 # ax mov         \ read function call
&> rdo-ram #) ds mov    \ point to rdo buffer seg

```


dos-call #) jmp

label set-dma

```
cli          \ disable interrupts
05 # al mov
al $0a # out  \ set mask on dma ch1
$45 # al mov
al 0c # out   \ clear byte pointer ff
al 0b # out   \ single mode,inrc,non auto,write transfer,ch1
```

dma-base #) ax mov

```
4 # cx mov here  ax shl  dl rcl  loop \ page to dl
$000f # dx and
0 [di] ax add    \ add write pointer offset
00 # dx adc      \ extend carry to page
al 02 # out
ah al mov
al 02 # out      \ output address to dma controller
dl al mov
al $83 # out     \ output page to dma controller
2 [di] ax mov
00 # ax cmp 0<>
if ax dec then
al 03 # out
ah al mov
al 03 # out      \ output 0 or n-1 word count to dma controller
sti            \ restart interrupts
01 # al mov
al $0a # out     \ reset mask dma ch1
ret
```

2variable rdo-dma-par \ ptr to buffer address and byte count

label install-rdo

```
cli
rdo-wrt #) ax mov
ax rdo-dma-par #) mov
dma-count # rdo-dma-par 2+ #) mov  \ setup rdo parameters
rdo-dma-par # di mov                \ point to rdo parameters
sti
do-poll-func #) rdo-input # cmp 0=
if
    set-dma #) jmp                  \ set up to receive rdo dma input
else
    rdo-handle #) bx mov
    set-cap #) jmp                  \ set up to receive rdo cap file input
then
```

```

label inc-rdo
  2 # &> tele-mode #) mov
  rdo-gauge #) dma-count # add      \ incr count
  rdo-wrt #) dma-count # add      \ incr pointer modulo rdo-size
  rdo-wrt #) rdo-size # cmp u>=
  if
    00 # rdo-wrt #) mov
  then
    rdo-cnt #) inc                  \ increment record count
  ret

```

variable tape-flg

```

label irq3-int (s -- )
  cli          \ suspend interrupts
  ax push
  bx push
  cx push
  dx push
  di push
  si push
  es push
  ds push
  bp push      \ save registers
  cs ax mov
  ax ds mov

  gse-stat # dx mov      \ point to gse status port
  dx al in              \ read status
  06 # dx add           \ point to interrupt flags
  ax push              \ save status
  dx push              \ save pointer

  $20 # al test 0<>     \ check for tape interrupt
  if
    al dx out          \ clear tape interrupt
    true # tape-flg #) mov \ set tape interrupt flag
  then
    dx pop             \ recover pointer
    ax pop             \ recover status

  08 # al test 0<>     \ check for dma terminal count interrupt
  if
    dx al in          \ clear terminal count interrupt
    inc-rdo #) call   \ point to new buffer space
    install-rdo #) call \ restart rdo dma
  then

  $20 # al mov
  al $20 # out \ end of interrupt to 8259

```

```

bp pop
ds pop
es pop
si pop
di pop
dx pop
cx pop
bx pop
ax pop \ restore registers
sti    \ enable interrupts
iret

```

```

code clr-interrupts (s -- )
  cli
  gse-stat 6 + # dx mov
  dx al in      \ clear tape int
  al dx out     \ clear dma t/c int
  sti
  next c;

```

only forth also editor also p_gse also hidden also

```

: install-dma (s -- )
  tape-flg off
  $0b get-int old-vec1 2!
  ?cs: irq3-int $0b set-intds
  rdo-wrt off          \ point to start of buffer
  %@> rdo-ram dma-base ! \ point to rdo buffer
inline
  cli
  install-rdo #) call
  gse-stat 6 + # dx mov
  dx al in          \ clear tape int
  al dx out         \ clear dma t/c int
  al $21 # in
  $ff # al xor
  $08 # al and
  al $21 # out
  sti
  next
end-inline
  stat-image @ dma-bit or tape-bit ff xor and stat! ;

```

only forth also editor also p_gse definitions

```

: clr-dma (s -- )
  stat-image @ dma-bit tape-bit and ff xor and stat!
  inline
  cli

```

```

05 # al mov
al $0a # out      \ set mask on dma ch1 dma
al $21 # in
$08 # al or
al $21 # out
sti
next
end-inline
old-vec1 2@ $0b set-intds ;

```

label save-rdo?

```
rdo-gauge #) rdo-size 2/ # cmp u>=
```

```
if
```

```
    rdo-gauge #) rdo-size 2/ # sub \ save 1/2 the buffer
```

```
    rdo-rd #) dx mov                \ point to data
```

```
    rdo-rd #) rdo-size 2/ # add
```

```
    rdo-rd #) rdo-size # cmp u>=
```

```
if
```

```
    00 # rdo-rd #) mov                \ point to next save modulo rdo buffer
```

```
then
```

```
    rdo-size 2/ # cx mov                \ get data size
```

```
    $4000 # ax mov                    \ write function call
```

```
    rdo-handle #) bx mov                \ get file handle
```

```
    &> rdo-ram #) ds mov                \ point to rdo buffer seg
```

```
    dos-call #) call
```

```
then
```

```
flush-rdo? #) true # cmp 0=
```

```
if
```

```
    rdo-gauge #) 00 # cmp 0<>
```

```
if
```

```
    rdo-gauge #) cx mov                \ get data size
```

```
    rdo-rd #) dx mov                \ point to data
```

```
    00 # rdo-gauge #) mov                \ clear gauge
```

```
    $4000 # ax mov                    \ write function call
```

```
    rdo-handle #) bx mov                \ get file handle
```

```
    &> rdo-ram #) ds mov                \ point to rdo buffer seg
```

```
    dos-call #) call
```

```
then
```

```
    >h close-rdo h>
```

```
then
```

```
ret
```

0 value cnt-2sec

variable 2sec-cntr

label 2sec-int (--)

```
cs: 2sec-cntr #) inc
```

```
cs: 2sec-cntr #) $1f # test 0=          \ 1.76 sec counter 0.569 hz.
```

```
if
```

```
    cs: 2sec-cnt #) inc
```

```

then
iret

```

```

0 value start-2sec

```

```

: set-2sec ( -- )
  start-2sec 0 =
  if
    $1c get-int old-vec2 2!
    ?cs: 2sec-int $1c set-intds
    2sec-cnt off
    2sec-cnt @
    begin
      dup 2sec-cnt @ <>
    until
    drop
    %on> start-2sec
  then ;

: clr-2sec ( -- )
  start-2sec 0 <>
  if
    old-vec2 2@ $1c set-intds
    %off> start-2sec
  then ;

```

```

defer real-time (S -- )

```

```

label real-poll ( -- )
  >h
    real-time
  h>
  ret

```

```

variable real-flg
  defer stop-tele

```

```

label rdo-poll (s -- )
  rdo-flg #) true # cmp 0=      \ and rdo flag is on
  if
    do-poll-func #) rdo-input # cmp 0=
    if
      save-rdo? #) call          \ rdo interface is the source
      rdo-cnt #) ax mov          \ check to save rdo capture data
      ax &> #rdo-rec #) mov
    else
      2sec-cnt #) 01 # test 0=   \ if rdo file is the source
      if
        rdo-cnt #) ax mov       \ and it is a 3.5 second count

```

```

    &> #rdo-rec #) ax cmp u<
    if
        install-rdo #) call      \ install for next read
        inc-rdo #) call          \ point to new data space
    else
        false # &> real-flg #) mov
        >h stop-tele h>
    then
    then
    then
    then
    ret

```

decimal

label b<->ang

```

-65.000 , , \ table end point
-63.759 , , -63.196 , , -62.640 , , -62.089 , ,
-61.543 , , -61.002 , , -60.465 , , -59.932 , ,
-59.403 , , -58.878 , , -58.355 , , -57.836 , ,
-57.320 , , -56.807 , , -56.296 , , -55.788 , ,
-55.282 , , -54.778 , , -54.276 , , -53.776 , ,
-53.277 , , -52.780 , , -52.285 , , -51.791 , ,
-51.298 , , -50.806 , , -50.316 , , -49.826 , ,
-49.337 , , -48.849 , , -48.362 , , -47.876 , ,
-47.390 , , -46.905 , , -46.421 , , -45.936 , ,
-45.453 , , -44.969 , , -44.486 , , -44.003 , ,
-43.520 , , -43.038 , , -42.556 , , -42.073 , ,
-41.591 , , -41.109 , , -40.626 , , -40.144 , ,
-39.661 , , -39.179 , , -38.696 , , -38.213 , ,
-37.729 , , -37.246 , , -36.762 , , -36.278 , ,
-35.794 , , -35.309 , , -34.824 , , -34.339 , ,
-33.853 , , -33.367 , , -32.880 , , -32.393 , ,
-31.906 , , -31.418 , , -30.930 , , -30.441 , ,
-29.952 , , -29.462 , , -28.971 , , -28.481 , ,
-27.989 , , -27.497 , , -27.005 , , -26.512 , ,
-26.019 , , -25.525 , , -25.030 , , -24.535 , ,
-24.039 , , -23.543 , , -23.047 , , -22.549 , ,
-22.052 , , -21.553 , , -21.054 , , -20.555 , ,
-20.055 , , -19.555 , , -19.054 , , -18.553 , ,
-18.051 , , -17.548 , , -17.045 , , -16.542 , ,
-16.038 , , -15.534 , , -15.029 , , -14.524 , ,
-14.019 , , -13.513 , , -13.006 , , -12.499 , ,
-11.992 , , -11.458 , , -10.977 , , -10.469 , ,
-09.960 , , -09.451 , , -08.942 , , -08.433 , ,
-07.923 , , -07.413 , , -06.903 , , -06.392 , ,
-05.882 , , -05.371 , , -04.860 , , -04.349 , ,
-03.838 , , -03.326 , , -02.815 , , -02.303 , ,
-01.791 , , -01.280 , , -00.768 , , -00.256 , ,

```

00.256 , , \ table end point

postfix

```
code (ang-tbl) ( n -- d )
  bx pop
  bx bx add
  bx bx add
  b<->ang 6 + [bx] dx mov
  b<->ang 4 + [bx] ax mov
  2push next c;
```

only forth also editor also p_gse also hidden also

```
: b->ang ( n -- d )
  255 and dup 0 127 between not
  if 255 swap - (ang-tbl) dabs
  else (ang-tbl)
  then ;
```

```
code ang->b ( d -- n )
  ax pcpx dx pop ax cx mov ax ax or 0>=
  if
    ax neg dx neg 0 # ax sbb
  then
    0 # di mov
    b<->ang 6 + [di] dx cmp u>= \ if on the table
    if
      -4 # di mov
      begin
        4 # di add dx b<->ang 6 + [di] cmp u>= \ Search for <=
      until
        di ax mov ax shr ax shr \ Calculate index

        b<->ang 6 + [di] bx mov
        b<->ang 2 + [di] bx add
        bx shr \ Calculate average
        dx bx cmp u>=
        if ax dec then \ Round up
      else
        0 # ax mov
      then
        cx cx or 0>= \ Was it positive ?
        if ax neg $ff # ax add then \ If positive find + number
        ax push
        next c; \ Push onto stack
```

```
label b->g-tbl
  0 c, 1 c, 3 c, 2 c, 6 c, 7 c, 5 c, 4 c,
  $c c, $d c, $f c, $e c, $a c, $b c, 9 c, 8 c,
```

```

code (b->g-tbl) ( n -- n )
  bx pop
  b->g-tbl [bx] ax mov
  ah ah sub
  lpush next c;

```

only forth also editor also p_gse also hidden also

```

: b->g ( b -- g )
  255 and $10 /mod swap over 01 and
  if 15 swap - then
  (b->g-tbl) swap
  (b->g-tbl) 16 * + ;

```

decimal

```

label g->b-tbl
  0 c, 1 c, 3 c, 2 c, 7 c, 6 c, 4 c, 5 c,
  $f c, $e c, $c c, $d c, 8 c, 9 c, $b c, $a c,

```

```

code (g->b-tbl) ( n -- n )
  bx pop
  g->b-tbl [bx] ax mov
  ah ah sub
  lpush next c;

```

only forth also editor also p_gse also hidden also

```

: g->b ( g -- b )
  255 and $10 /mod
  (g->b-tbl) swap
  (g->b-tbl) over 01 and
  if 15 swap - then swap 16 * + ;

```

64 value #steps

comment:

At 18.2 calls/sec, 2sec-int divides by 32 to increment 2sec-cnt. Change in 2sec-cnt triggers ssh-poll (32 cnts/18.2 cnts/sec = 1.758 sec). The variable ssh-cntr is incremented at this rate (0.569 hz). When ssh-cntr is divided by ssh-xn (ssh-yn) the ssh-xstep (ssh-ystep) rate is determined.

The value #steps is the number of steps in the cosine table. This means that each cycle takes #steps * ssh-xstep period (ssh-ystep) 360 deg/cy / (64 steps/cy * ssh-xn cnt/steps * 1.758 sec/cnt) deg/sec 3.2 deg/sec / ssh-n 3.2/60 deg/min / ssh-n -90 deg/min <= w0 <= 90 deg/min

ssh-n = 32 * 600 / w0

comment;

only forth also editor also p_gse also hidden also

```
: ang ( -- )
  create
  floating pi 2.0e0 f* #steps 1- 0 float f/
  #steps 0
  do
    fdup i 0 float f* fcos 1000. float f* fix , ,
  loop fdrop
;
```

ang cos-tbl

```
code ang-tbl ( n -- d )
  bx pop
  #steps 1- # bx and
  bx bx add
  bx bx add
  cos-tbl 2 + [bx] dx mov
  cos-tbl 0 + [bx] ax mov
  2push next c;
```

only forth also editor also p_gse also hidden also

```
0 value ssh-cntr

4 value ssh-xn
0 value ssh-xstep
-10.000 ang->b b->g \ Default for test mode -10 deg
  value ssh-xsig
90 value ssh-xw0
40 value ssh-x0

4 value ssh-yn
0 value ssh-ystep
10.000 ang->b b->g \ Default for test mode 10 deg
  value ssh-ysig
90 value ssh-yw0
20 value ssh-y0

-1 value ssh-flg \ Default non test
-1 value ssh-ata-flg \ Default ata on
```

```
0 constant ssh-x
1 constant ssh-y
2 constant ssh-ata
```

```

0 value ssh-xg
0 value ssh-yg
0 value ssh-xb
0 value ssh-yb

```

comment:

```

: ssh! ( -- )
  0 20 at
  ssh-ata-flg 5 u.hr
  ssh-xg $ff and 5 u.hr
  ssh-yg $ff and 5 u.hr
  0 21 at
    $b 5 u.hr
  ssh-xb $ff and 5 u.hr
  ssh-yb $ff and 5 u.hr ;

```

comment;

```

: ssh! ( -- )
  ssh-xg ssh-yg ssh-ata-flg
  ssh-ata dc!
  ssy-fix ssh-y dc! \ Fix for swaped nibbles
  ssh-x dc! ;

```

decimal

```

: amp->g (s dacos,amp --a,g)
>r if negate then r> *d
ang->b dup b->g ;

```

```

: ssh-xcall ( -- )
  ssh-xn 0= ssh-x0 0= or not
  if
    ssh-cntr ssh-xn mod 0 =
    if
      %incr> ssh-xstep \ incr cos step
      ssh-xstep #steps 1- and \ get step count
      ang-tbl ssh-x0 amp->g
      %!> ssh-xg
      %!> ssh-xb
    then
  then ;

```

```

: ssh-ycall ( -- )
  ssh-yn 0= ssh-y0 0= or not
  if
    ssh-cntr ssh-yn mod 0 =
    if
      %incr> ssh-ystep \ incr cos step
      ssh-ystep #steps 1- and \ get step count

```

```

        ang-tbl ssh-y0 amp->g
        %!> ssh-yg
        %!> ssh-yb
    then
then ;

```

```

: (ssh-poll) ( -- )
  ssh-flg
  if
    %incr> ssh-cntr
    ssh-xcall
    ssh-ycall
  else
    ssh-x0 %!> ssh-xg
    ssh-y0 %!> ssh-yg
  then
  ssh! ;

```

```

label ssh-poll (S -- )
>h  ssh-x0 %!> ssh-xg
    ssh-y0 %!> ssh-yg
    ssh!
h>  ret

```

```

\  >h (ssh-poll) h> ret

```

only forth also editor also p_gse also hidden also

```

: vu-rdo (s n -- )
  $200 * rdo-ram swap $20 ldup ;

```

```

0 value parx
0 value pary
-1 value dbuf-close-flg

```

```

: par-pos (s n -- )
  pary + parx swap at ;

```

```

: nam-pos (s n -- )
  pary + parx 11 - swap at ;

```

```

: msg-pos (s n -- )
  pary + parx 6 + swap at ;

```

```

: .par-stat
  view-flg
  if
    $a 0 swap 20 over 13 +
    2over %!> pary 14 + %!> parx box&fill
  then

```

```

." dma word cnt      " bcr
." dma addr cnt      " bcr
." dma status        " bcr
." rdo-wrt           " bcr
." rdo-gauge         " bcr
." rdo-rd            " bcr
." cnt-2sec          " bcr
." rdo-flg           " bcr
." real-flg          " bcr
." dbuf-close-flg    " bcr
." #rdo-rec          " bcr
." #dbuf-rec         " bcr
00 $c pc!
03 pc@ 03 pc@ $100 * + 01 par-pos 5 u.hr
02 pc@ 02 pc@ $100 * + 02 par-pos 5 u.hr
08 pc@ $0f and      03 par-pos 5 u.hr
rdo-wrt @           04 par-pos 5 u.hr
rdo-gauge @         05 par-pos 5 u.hr
rdo-rd @            06 par-pos 5 u.hr
cnt-2sec            07 par-pos 5 u.hr
rdo-flg @           08 par-pos 5 u.hr
real-flg @          09 par-pos 5 u.hr
dbuf-close-flg     10 par-pos 5 u.hr
#rdo-rec            11 par-pos 5 u.hr
#dbuf-rec           12 par-pos 5 u.hr

```

then ;

```

label sys-stat (s -- )
  >h .par-stat h>
  ret

```

\ ibm interface board support - background file saver 2sep89cdb

only forth also editor also p_gse also hidden also

```

code (do-poll) ( -- )
  2sec-cnt #) ax mov          \ check 2sec count
  &> cnt-2sec #) ax cmp 0<>
  if                          \ if changed
    ax &> cnt-2sec #) mov      \ save change
    ax push
    bx push
    cx push
    dx push
    rdo-poll #) call
    real-poll #) call          \ check real time display routines

```

```

    ssh-poll #) call
    sys-stat #) call
    dx pop
    cx pop
    bx pop
    ax pop
  then
  next c;

```

```

background: do-poll (s -- )
begin pause (do-poll) again ;

```

only forth also editor also p_gse also hidden also

```

: clr-cap-out (s -- )
  flush-rdo? on
  begin pause flush-rdo? @ not key? or until
  rdo-flg off ;

```

\ ibm interface board support - capture file (input) 21sep89cdb
hex

```

code +rdo/mod (s addr,n -- addr+n)          \ Sum mod rdo
  cx pop
  ax pop
  cx ax add
  rdo-size # ax cmp u<=
  if
    rdo-size # ax sub
  then
    00 # ax cmp u>
  if
    rdo-size # ax add
  then
  ax push
next c;

```

```

code -rdo/mod (s addr1,addr2 -- n)  \ addr1 < addr2 mod rdo
  cx pop                               \ Subtract mod rdo
  ax pop
  ax cx sub 0<
  if
    rdo-size # cx add
  then
  cx push
next c;

```

only forth also p_gse definitions

```

: rdo-@ (s addr -- n)          \ Read word mod rdo
  dup 1 +rdo/mod rdo-c@ $100 * swap rdo-c@ + ;

: 0e8e? (s -- addr,len/flg)
  rdo-ptr @ dup rdo-wrt @ -rdo/mod \ Get addr and length
  dup $200 u>=
  if \ Enough data to search
    2dup 0
    ?do
      dup rdo-c@ $0e =
      if
        dup rdo-@ $8e0e =
        if
          rot drop swap i - leave \ If found adj addr and length
        then
      then
        1 +rdo/mod
        over 1- i =
        if
          >r 2drop r> false \ if not found return addr,false
        then
      loop
    then ;

```

```

$1e2 constant data-bytes
$2 constant rec-offset
$18 constant buf-offset
$8 constant mclk-offset
$10 constant sso-offset
$14 constant ancil-offset
$1c constant data-offset
$1fe constant crc-offset
$8e0e constant fr-head

```

```

label 1-bit
dx bx mov
dx shl
ah al and

```

```

00 # al cmp 0<>
if
  dx inc
then

```

```

$8000 # bx test 0<>
if
  $1021 # dx xor
then
  ret

```

```

code crc-calc (s old-crc,byte -- new-crc)

```

```

ax pop
cx pop
bx push
cx dx mov
$80 # ah mov
# 8 cx mov here
  ax push
  1-bit #) call
  ax pop
  ah shr
loop
bx pop
dx push
next c;

```

```

: chk-crc (s addr -- flg )           \ Check record crc
  crc-offset over
+rdo/mod rdo-@
  swap 0 crc-offset 0
  do
    over i +rdo/mod rdo-c@ crc-calc
  loop
  0 crc-calc 0 crc-calc nip = ;

```

```

0 value rdo-rec
0 value rdo-crc-flg

0 value rdo-mclk0
0 value rdo-mclk1
0 value rdo-mclk2
0 value rdo-sso
0 value rdo-sso-ata
0 value rdo-ancil

```

```

0 value buf-ptr
0 value buf-page
0 value bad-crcs
0 value good-crcs
0 value rdo-addr

```

```

0 value buf-cnt
0 value delta-rec
6 data-bytes *
  value dacs-buf-size

```

```

defer buf ( -- )

```

```

: srch-rec (s -- flg )
  0e8e?

```

\ Search for

```

header
  $200 u>=
  if
data
  rdo-rec >r
  dup rec-offset      +rdo/mod rdo-@
  flip $3fff and dup      %!> rdo-rec      \ Get rec #
  r> - %!> delta-rec
\  dup mclk-offset      +rdo/mod rdo-@      %!> rdo-mclk0      \ Get mclk data
\  dup mclk-offset 2 + +rdo/mod rdo-@      %!> rdo-mclk1      \ Get mclk data
\  dup mclk-offset 4 + +rdo/mod rdo-@      %!> rdo-mclk2      \ Get mclk data
  dup sso-offset      +rdo/mod rdo-@      %!> rdo-sso      \ Get sso data
  dup sso-offset 2 + +rdo/mod rdo-@
  $0100 and
  if
    %off> rdo-sso-ata
  else
    %on> rdo-sso-ata
  then
data
  flg
  dup ancil-offset      +rdo/mod rdo-@      %!> rdo-ancil      \ Get ancil
  buf-ptr $7fff and buf-page >r >r
  dup buf-offset      +rdo/mod rdo-@      %!> buf-ptr      \ Get buf #
  dup buf-offset 2 + +rdo/mod rdo-@      %!> buf-page      \ Get page #
  buf-ptr $7fff and r> -
  buf-page r> - dacs-buf-size *
  + %!> buf-cnt
  dup chk-crc
  dup %!> rdo-addr
  $200 +rdo/mod rdo-ptr !
  rdo-crc-flg not
  if
    beep
    %incr> bad-crcs
  else
    %incr> good-crcs
  then
  buf
  then
  else
  rdo-ptr !
  then ;
\  Move data to dbuf
\  Save search pointer

dacs-buf-size
  value dbuf-size
0 value dbuf-ram
variable dbuf-wrt
variable dbuf-rd
variable dbuf-gauge
2variable dbuf-ptr

```


2variable w-ptr

```
: init-dbuf (s -- )
  &> dbuf-ram dbuf-size alloc-mem
  dbuf-wrt    off
  dbuf-rd     off
  dbuf-gauge  off
  %off> bad-crcs
  %off> good-crcs
  %off> buf-cnt
  %off> delta-rec
  0 0 dbuf-ptr 2!
  0 0 w-ptr 2!
  dbuf-ram
  0 dbuf-size bounds
  do $ff over i c!l loop drop ;
```

```
code dbuf-seg ( -- addr )
  &> dbuf-ram #) 0 # cmp 0=
  if
    >h
    beep ." dbuf not allocated"
    init-dbuf
    h>
  then
  &> dbuf-ram #) push
  next c;
```

```
: dbuf-c! ( n,addr -- )
  dbuf-seg swap c!l ;
```

```
: dbuf-! ( n,addr -- )
  dbuf-seg swap !l ;
```

```
: dbuf-c@ ( addr -- n )
  dbuf-seg swap c@l ;
```

```
: dbuf-@ ( addr -- n )
  dbuf-seg swap @l ;
```

```
variable dbuf-handle
0 value dbuf-h-addr
-1 value new-dbuf-page?
```

```
: ?save-d (s -- )
  saveb-flg
  if
```

```

new-dbuf-page?
  if
    dbuf-seg 0 dbuf-size dbuf-h-addr write-file
    %off> new-dbuf-page?
  then
then ;

: make-dbuf (s -- / create rdo-00.buf file)
  &> dbuf-h-addr " rdo-00.buf " ">$ make-file
  dbuf-handle ! ;

0 value e-page
0 value e-ptr

: clr-dbuf-out ( -- )
  saveb-flg
  if
    w-ptr 2+ @ 0 <>
    if
      dbuf-seg 0 w-ptr 2 + @ dbuf-h-addr write-file
    then
      dbuf-h-addr endfile %!> e-page %!> e-ptr
      dbuf-h-addr hclose %!> dbuf-close-flg
    then ;

: ptr- (s ptr1,pg1,ptr2,pg2, -- n ) \ 1 > 2
  rot swap - dbuf-size * -rot - + ;

: wrt-d (s n -- )
  w-ptr 2+ @ tuck dbuf-c!
  1+ dup dbuf-size u< not
  if
    drop 0
    %incr> w-ptr
    ?save-d
  then
  w-ptr 2+ ! ;

: save-data (s ptr,pg -- ) \ ptr < d-ptr+data-bytes
  2dup
  dbuf-ptr 2! \ Save pointer to last data saved in dbuf
  w-ptr 2! \ Point to
  rdo-addr data-offset +rdo/mod \ Point to rdo data
  data-bytes 0
  ?do
    dup rdo-c@
    wrt-d
    1 +rdo/mod
  loop drop ;

```

```

: ptr-mark (s ptr,pg -- ) \ ptr > d-ptr+data-bytes
" {{{ " bounds do i c@ wrt-d loop
0 (ud.) bounds do i c@ wrt-d loop
0 (ud.) bounds do i c@ wrt-d loop
" }}} " bounds do i c@ wrt-d loop ;

: new-ptr (s ptr,pg -- )
4 0 do wrt-d loop \ Save w-ptr,new-ptr
2dup
w-ptr 2@ drop over \ Adjust w-ptr pointer to new page
d> \ If new page < w-ptr
if 1 + then \ move w-ptr to next page
w-ptr 2! \ Update w-ptr
save-fill ; \ Fill to new pointer

: d-between (s d#,d-min,d-max -- flg)
>r >r 2over d> -rot r> r> d> or not ;

0 value old-ptr
0 value old-page
0 value cur-ptr
0 value cur-page

: ptr-step? ( ptr,page -- flg)
dbuf-ptr 2@ w-ptr 2@ d-between not ;

: t-ptr->d-ptr ( ptr,pg -- addr )
cur-page + swap cur-ptr + ;

: d-ptr->t-ptr ( addr -- ptr,pg )
swap cur-ptr + ;

: (buf) ( -- )
buf-ptr buf-page
swap $7fff and swap
2dup ptr-step?
if
ptr-mark
new-ptr
else
save-d?
then

good-crchs 1 =
if
2dup
dbuf-ptr 2!

```

```

    0 over
    w-ptr 2!
    %on> new-dbuf-page?
  then

  2dup
  dbuf-ptr-chk
  if
    save-data
  else
    2drop
  then ;

```

```

' (buf) is buf

```

```

code +d/mod (s addr,n -- addr+n)
  cx pop
  ax pop
  cx ax add
  dbuf-size # ax cmp u<=
  if
    dbuf-size # ax sub
  then
    0 # ax cmp u>
  if
    dbuf-size # ax add
  then
    ax push
  next c;

```

```

code -d/mod (s addr1,addr2 -- n) \ addr1 < addr2 mod d
  cx pop
  ax pop
  ax cx sub 0<
  if
    dbuf-size # cx add
  then
    cx push
  next c;

```

```

: ch= (s ch1,ch2 -- flg)
  upc swap upc = ;

```

```

: ch@ (s addr -- ch)
  dbuf-c@ $7f and ;

```

```

: srch-d (s addr,len,baddr,blen -- addr',len',flg)
3 pick ch@ -rot 0
do
  2dup ch@ ch=
  if 2swap true 4dup drop 0
  do
    2dup i + ch@ swap i +d/mod dbuf-c@ ch= 3 roll and -rot
  loop
  2drop >r 2swap r> dup
  if
    leave
  else
    drop 1 +d/mod
  then
  else
    1 +d/mod
  then
loop
dup true = not
if
  0
then
rot drop 2swap 2drop 4dup drop rot swap -d/mod - swap ;

```

```

0 value res>
0 value res-name>
0 value dog>
0 value dog-name>

```

```

0 value or*>
0 value or*-name>
0 value or1>
0 value or1-name>
0 value or2>
0 value or2-name>
0 value or3>
0 value or3-name>
0 value or4>
0 value or4-name>
0 value or5>
0 value or5-name>
0 value or6>
0 value or6-name>

```

```

0 value se*>
0 value se*-name>
0 value se1>
0 value se1-name>
0 value se2>
0 value se2-name>
0 value se3>

```

0 value se3-name>

0 value el*>
0 value el*-name>
0 value el1>
0 value el1-name>
0 value el2>
0 value el2-name>
0 value el3>
0 value el3-name>
0 value el4>
0 value el4-name>

0 value stb>
0 value stb-name>
0 value gar>
0 value gar-name>
0 value occ>
0 value occ-name>
0 value hsk>
0 value hsk-name>
0 value tpm>
0 value tpm-name>
0 value lmp>
0 value lmp-name>
0 value hvs>
0 value hvs-name>

0 value hv2>
0 value hv2-name>
0 value tss>
0 value tss-name>
0 value tst>
0 value tst-name>
0 value cal>
0 value cal-name>
0 value tmk>
0 value tmk-name>

0 value iv*>
0 value iv*-name>
0 value iv1>
0 value iv1-name>
0 value iv2>
0 value iv2-name>
0 value iv3>
0 value iv3-name>
0 value iv4>
0 value iv4-name>
0 value iv5>
0 value iv5-name>
0 value iv6>

```

0 value iv6-name>

0 value blank-name>
0 value data1>
0 value data2>

here "                " !> blank-name>

blank-name> value data1-name>
blank-name> value data2-name>

here " res" !> res-name>
here " dog" !> dog-name>

here " or" !> or*-name>
here " or1" !> or1-name>
here " or2" !> or2-name>
here " or3" !> or3-name>
here " or4" !> or4-name>
here " or5" !> or5-name>
here " or6" !> or6-name>

here " se" !> se*-name>
here " se1" !> se1-name>
here " se2" !> se2-name>
here " se3" !> se3-name>

here " el" !> el*-name>
here " el1" !> el1-name>
here " el2" !> el2-name>
here " el3" !> el3-name>
here " el4" !> el4-name>

here " stb" !> stb-name>
here " gar" !> gar-name>
here " occ" !> occ-name>
here " hsk" !> hsk-name>
here " tpm" !> tpm-name>
here " lmp" !> lmp-name>
here " hvs" !> hvs-name>
here " hv2" !> hv2-name>
here " tss" !> tss-name>
here " tst" !> tst-name>
here " cal" !> cal-name>
here " tmk" !> tmk-name>

here " iv" !> iv*-name>
here " iv1" !> iv1-name>
here " iv2" !> iv2-name>
here " iv3" !> iv3-name>
here " iv4" !> iv4-name>
here " iv5" !> iv5-name>

```

here " iv6" !> iv6-name>

label data>-tbl \ data-tbl msg>, handler>

here &> res-name> , ' noop , !> res>

here &> dog-name> , ' noop , !> dog>

here &> or*-name> , ' noop , !> or*>

here &> or1-name> , ' noop , !> or1>

here &> or2-name> , ' noop , !> or2>

here &> or3-name> , ' noop , !> or3>

here &> or4-name> , ' noop , !> or4>

here &> or5-name> , ' noop , !> or5>

here &> or6-name> , ' noop , !> or6>

here &> se*-name> , ' noop , !> se*>

here &> se1-name> , ' noop , !> se1>

here &> se2-name> , ' noop , !> se2>

here &> se3-name> , ' noop , !> se3>

here &> el*-name> , ' noop , !> el*>

here &> el1-name> , ' noop , !> el1>

here &> el2-name> , ' noop , !> el2>

here &> el3-name> , ' noop , !> el3>

here &> el4-name> , ' noop , !> el4>

here &> stb-name> , ' noop , !> stb>

here &> gar-name> , ' noop , !> gar>

here &> occ-name> , ' noop , !> occ>

here &> hsk-name> , ' noop , !> hsk>

here &> tpm-name> , ' noop , !> tpm>

here &> lmp-name> , ' noop , !> lmp>

here &> hvs-name> , ' noop , !> hvs>

here &> hv2-name> , ' noop , !> hv2>

here &> tss-name> , ' noop , !> tss>

here &> tst-name> , ' noop , !> tst>

here &> cal-name> , ' noop , !> cal>

here &> tmk-name> , ' noop , !> tmk>

here &> iv*-name> , ' noop , !> iv*>

here &> iv1-name> , ' noop , !> iv1>

here &> iv2-name> , ' noop , !> iv2>

here &> iv3-name> , ' noop , !> iv3>

here &> iv4-name> , ' noop , !> iv4>

here &> iv5-name> , ' noop , !> iv5>

here &> iv6-name> , ' noop , !> iv6>

here " s depth " !> data1-name>
&> e-page !> data1>


```

here " r depth          " !> data2-name>
&> e-ptr    !> data2>

```

decimal

```

: (real-time) (s -- )
  real-flg
  if
    srch-rec
  then
    view-flg
    if \ up lft    width    length
      40 $c over 23 + over 10 +
      2over %!> pary 12 + %!> parx box&fill

      ." tele rec #          " bcr
      ." bad crc cnt        " bcr
      ." buf-page           " bcr
      ." buf-ptr            " bcr
      ." sso hardware       " bcr
      ." sso telemetry      " bcr
      ." ---data window---  " bcr
      ."                    " bcr
      ."                    "

      rdo-rec      01 par-pos 5 u.hr
      rdo-crc-flg  02 msg-pos
      if ." Good"
      else ." Bad "
      then
      bad-crcs     02 par-pos 5 u.hr
      buf-page     03 par-pos 5 u.hr
      buf-ptr      $7fff and
                  04 par-pos 5 u.hr
      buf-ptr      04 msg-pos
      $8000 and
      if ." Fill"
      else ." Full"
      then
      sso@ 01 and   05 msg-pos
      if ." Off"
      else ." On "
      then
      swap join    05 par-pos 5 u.hr
      rdo-sso sso-fix
                  06 par-pos 5 u.hr
      rdo-sso-ata  06 msg-pos
      if ." On "
      else ." Off"
      then
      \      datal-name>    08 nam-pos count type

```

```

\          data1> @      08 par-pos 5 u.hr
\          data2-name>   09 nam-pos count type
\          data2> @      09 par-pos 5 u.hr
          data1-name>    08 nam-pos count type
          depth          08 par-pos . .s
          data2-name>    09 nam-pos count type
          rp0 @ rp@ - 2/  09 par-pos . .s
then ;

: ty (real-time) ;
\ : ty dark %on> view-flg (real-time) ;

' (real-time) is real-time

: (tele-select) ( -- )
  init-rdo
  source-flg 0 =
  if
    clr-interrupts
    rdo-input do-poll-func !      \ set for rdo input
    install-dma
  else
    rdo-input not do-poll-func !
    open-rdo
  then

  save-flg
  if
    rdo-input do-poll-func !      \ set for rdo input
    make-rdo
  then
  init-dbuf
  saveb-flg
  if
    %on>   dbuf-close-flg
    %off> #dbuf-rec
    make-dbuf
  then

  real-flg on
  1 %!> tele-mode ;

: (tele-stop) ( -- )
  real-flg off
  saveb-flg
  if
    clr-dbuf-out
  then
  save-flg
  if
    clr-cap-out

```

then

```
source-flg 0 =  
if clr-dma else close-rdo then  
&> dbuf-ram dealloc-mem  
&> rdo-ram dealloc-mem  
0 %!> tele-mode ;
```

' (tele-stop) is stop-tele

only forth also editor also p_gse definitions

```
'0' value dacs-chan  
0 value pos<-dacs-cal
```

```
: set<-dacs-cal ( n -- )  
%!> pos<-dacs-cal  
4 7 bounds  
do  
  box-corner 2@ 17 i d+ at  
  pos<-dacs-cal i = if " <- " else " " then type  
loop ;
```

```
: .dacs-chan ( -- )  
  box-corner 2@ 22 3 d+ at dacs-chan emit  
  pos<-dacs-cal set<-dacs-cal ;
```

```
: dacs-select ( n -- )  
%!> dacs-chan .dacs-chan ;
```

```
: dacs-cal
```

```
  savescr  
  ['] noop save!> dobutton  
  ?doingmac 0= \ if we are doing a macro, don't display  
                \ command menu box.  
  if screenline 1+ dup 12 >  
    if 13 -  
    then 18 swap 58 over 14 +  
          2over box-corner 2!  
          box&fill  
  
    ."          dacs cal menu          key " bcr bcr  
    ." select cal channel #          0-7 " bcr  
    ." cal ad1 mux a          alt-f1 " bcr  
    ." cal ad1 mux b          alt-f2 " bcr  
    ." cal ad2 mux a          alt-f3 " bcr  
    ." cal ad2 mux b          alt-f4 " bcr
```

```

        ."    cal Rtd                                R " bcr
        ."    cal Dacs pmon                           D " bcr
        ."    cal Asis pmon                           A " bcr
        ."    cal File menu                           F " bcr
        ."    Quit                                    Q " bcr
        ."    \s10\r esc \0 = cancel"
        .dacs-chan
    then
    begin
        key bl or >r
        r@ dup '0' '7' between
            if dacs-select
            else drop then
                $e8 r@ = if ad1a-tbl 4 set<-dacs-cal then
                $e9 r@ = if ad1b-tbl 5 set<-dacs-cal then
                $ea r@ = if ad2a-tbl 6 set<-dacs-cal then
                $eb r@ = if ad2b-tbl 7 set<-dacs-cal then
                'r' r@ = if rtd-tbl 8 set<-dacs-cal then
                'd' r@ = if dacs-ptbl 9 set<-dacs-cal then
                'a' r@ = if asis-ptbl $a set<-dacs-cal then
                'f' r@ = if cal-files $b set<-dacs-cal then
                59 r@ = if menu-exit r>drop 113 >r then
            r> 113 =
        until
        restscr
        restore> dobutton
        sdisplay showstat cursor-on ;

```

```

: .flg ( flg -- )
  if ." ON" else ." OFF" then ;

: .source-flg ( -- )
  box-corner 2@ 20 3 d+ at source-flg
  if ." CAP " else ." RDO " then ;

: .save-flg ( -- )
  box-corner 2@ 20 4 d+ at save-flg .flg ;

: .saveb-flg ( -- )
  box-corner 2@ 20 5 d+ at saveb-flg .flg ;

: .view-flg ( -- )
  box-corner 2@ 20 6 d+ at view-flg .flg ;

: .tele-flg ( -- )
  box-corner 2@ 20 7 d+ at tele-mode
  dup 0 = if ." OFF " then
  dup 1 = if ." WAITING" then
  2 = if ." RUNNING" then ;

: .tele-mode ( -- )

```

```

        .source-flg  .save-flg  .saveb-flg .view-flg  .tele-flg ;

: source-select ( -- )
  source-flg
  if
    %off> source-flg  %on> save-flg
  else
    %on> source-flg  %off> save-flg
  then ;

: save-select ( -- )
  save-flg
  if %off> save-flg
  else
    source-flg
    if else
      %on> save-flg  \ on if rdo selected
    then
  then ;

: saveb-select ( -- )
  saveb-flg if %off> saveb-flg else %on> saveb-flg then ;

: view-select ( -- )
  view-flg
  if
    restscr
    savescr
    %off> view-flg
    restscr
  else
    %on> view-flg
    savescr
  then ;

: view-check-in ( -- )
  view-flg
  if
    savescr
  then ;

: view-check-out ( -- )
  view-flg
  if
    restscr
  then ;

: tele-select ( -- )
  (tele-select) ;

```

```
: tele-stop ( -- )
  (tele-stop)
  box-corner 2@ 20 8 d+ at "      " type ;
```

```
: tele-menu
```

```
  savescr
  ['] noop save!> dobutton
  ?doingmac 0=      \ if we are doing a macro, don't display
                  \ command menu box.
  if      screenline 1+ dup 12 >
    if      13 -
    then    18 swap 58 over 11 +
            2over box-corner 2!
            box&fill

    ."      telemetry menu          key " bcr bcr
    ." Telemetry source      (toggle) T " bcr
    ." save Cap file        (toggle) C " bcr
    ." Save buf file        (toggle) S " bcr
    ." View telemetry        (toggle) V " bcr
    ." Begin telemetry      B " bcr
    ." End telemetry        E " bcr
    ." Quit                  Q " bcr
    ." \s10\r esc \0 = cancel"
    cursor-off
    view-check-in
  then
  begin
    begin
      .tele-mode pause key?
    until
      key bl or >r
      tele-mode 0 =
      if
        't' r@ = if source-select then
        'b' r@ = if tele-select then
        'c' r@ = if save-select then
        's' r@ = if saveb-select then
      else
        't' r@ = 'b' r@ = 'c' r@ = 's' r@ = or or or
        if
          box-corner 2@ 20 8 d+ at " must end" type
          then
            'e' r@ = if tele-stop then
          then
            'v' r@ = if view-select then
            59 r@ = if menu-exit r>drop 113 >r then
            .tele-mode
            r> 113 =
          until
```

```

view-check-out
restscr
restore> dobutton
sdisplay showstat cursor-on ;

```

```

variable ssh-buff 3 allot

```

```

: end-input ( -- angle,flg)
  ssh-buff 5 blank
  ssh-buff 1+ 3 expect span @ ssh-buff c!
  ssh-buff number? nip ;

```

```

: input-angle ( addr --- )
  savescr cursor-off
  ['] noop save!> dobutton
  14 6 70 11 box&fill
  ." input error angle amplitude in degrees (+/- 90 max)" bcr
  ." \s10press enter \r key \0 to continue. " bcr
  ." \s20 "
  end-input
  if swap !
  else 2drop
  then
  restore> dobutton
  restscr cursor-on ;

```

```

: input-rate ( addr --- )
  savescr cursor-off
  ['] noop save!> dobutton
  14 6 70 11 box&fill
  ." input error angle rate in degrees/min (+/- 90 max) " bcr
  ." \s10press enter \r key \0 to continue. " bcr
  ." \s20 "
  end-input
  if swap !
  else 2drop
  then
  restore> dobutton
  restscr cursor-on ;

```

```

: .ssh-x0 ( -- )
  box-corner 2@ 20 3 d+ at ssh-x0 3 .r ;

```

```

: set-ssh-x0 ( -- )
  %&> ssh-x0 input-angle .ssh-x0 ;

```

```

: .ssh-xw0 ( -- )

```

```

box-corner 2@ 20 4 d+ at ssh-xw0 3 .r ;

: set-ssh-xw0 ( -- )
  %&> ssh-xw0 input-angle
  ssh-xw0
  if
    32 600 ssh-xw0 */ %!> ssh-xn
  else
    %off> ssh-xn
  then
    .ssh-xw0 ;

: .ssh-y0 ( -- )
  box-corner 2@ 20 5 d+ at ssh-y0 3 .r ;

: set-ssh-y0 ( -- )
  %&> ssh-y0 input-angle .ssh-y0 ;

: .ssh-yw0 ( -- )
  box-corner 2@ 20 6 d+ at ssh-yw0 3 .r ;

: set-ssh-yw0 ( -- )
  %&> ssh-yw0 input-angle
  ssh-yw0
  if
    32 600 ssh-yw0 */ %!> ssh-yn
  else
    %off> ssh-yn
  then
    .ssh-yw0 ;

: .ssh-ata-flg ( -- )
  box-corner 2@ 20 7 d+ at ssh-ata-flg .flg ;

: togg-ata ( -- )
  ssh-ata-flg
  if %off> ssh-ata-flg else %on> ssh-ata-flg then
    .ssh-ata-flg ;

: .ssh-flg ( -- )
  box-corner 2@ 20 8 d+ at ssh-flg not .flg ;

: ssh-mode ( -- )
  ssh-flg
  if %off> ssh-flg else %on> ssh-flg then
    .ssh-flg ;

: .ssh ( -- )
  .ssh-x0 .ssh-xw0 .ssh-y0 .ssh-yw0 .ssh-ata-flg .ssh-flg ;

```



```

: init-ssh ( -- )
  32 600 ssh-xw0 */ %!> ssh-xn
  32 600 ssh-yw0 */ %!> ssh-yn ;

: ssh-menu

  savescr
  ['] noop save!> dobutton
  ?doingmac 0= \ if we are doing a macro, don't display
               \ command menu box.
  if          screenline 1+ dup 12 >
    if        13 -
    then      18 swap 58 over 11 +
              2over box-corner 2!
              box&fill

              ."          ss simulator menu          key " bcr bcr
              ." 1 set ssh x0          degrees      1 " bcr
              ." 2 set ssh xw0         deg/sec      2 " bcr
              ." 3 set ssh y0          degrees      3 " bcr
              ." 4 set ssh yw0         deg/sec      4 " bcr
              ." toggle ssh Ata        (toggle)     A " bcr
              ." ssh Test mode        (toggle)     T " bcr
              ." Quit                  (toggle)     Q " bcr
              ." \s10\r esc \0 = cancel"
              .ssh
  then
  begin
    key bl or >r
    '1' r@ = if set-ssh-x0 then
    '2' r@ = if set-ssh-xw0 then
    '3' r@ = if set-ssh-y0 then
    '4' r@ = if set-ssh-yw0 then
    'a' r@ = if togg-ata then
    't' r@ = if ssh-mode then
    59 r@ = if menu-exit r>drop 113 >r then
    r> 113 =
  until
  restscr
  restore> dobutton
  sdisplay showstat cursor-on ;

newmenu gse-menu
  menuline"      pasp gse menu          key " noop
  \u ops-menu    menuline" Operations menu      O " ops-menu
  \u dev-menu    menuline" Development menu     D " dev-menu
  \u cal-menu    menuline" Calibrations menu    C " cal-menu
  \u setup-menu  menuline" Setup menu           S " setup-menu
  menuline" Quit          Q " noop

```

endmenu

newmenu ops-menu

	menuline"	operations menu	key "	noop
\u dacs-cmds	menuline"	Dacs commands menu	D "	dacs-cmds
\u test-cmds	menuline"	Gse test menu	G "	test-cmds
\u tpm-cmds	menuline"	Send tpm commands menu	S "	tpm-cmds
\u cal-cmds	menuline"	Cal commands menu	C "	cal-cmds
\u tele-menu	menuline"	Telemetry menu	T "	tele-menu
	menuline"	Quit	Q "	noop

endmenu

newmenu dev-menu

	menuline"	development menu	key "	noop
\u dacs-meta	menuline"	Download pasp.blk	D "	dacs-meta
\u make-pasp	menuline"	make new Flight roms	F "	make-pasp
\u make-dev	menuline"	make new development Roms	R "	make-dev
\u terminal	menuline"	Terminal	T "	terminal
\u hello	menuline"	Go to forth	G "	hello
	menuline"	Quit	Q "	noop

endmenu

newmenu cal-menu

	menuline"	calibrations menu	key "	noop
\u asis-cal	menuline"	Asis cal menu	A "	asis-cal
\u dacs-cal	menuline"	Dacs cal menu	D "	dacs-cal
\u gfe-cal	menuline"	Gfe cal menu	G "	gfe-cal
\u cal-files	menuline"	cal File menu	f "	cal-files
	menuline"	Quit	Q "	noop

endmenu

newmenu setup-menu

	menuline"	setup menu	key "	noop
\u tele-menu	menuline"	Telemetry menu	T "	tele-menu
\u ssh-menu	menuline"	Sun senor simulator	S "	ssh-menu
\u real-setup	menuline"	Real time display setup	R "	real-setup
	menuline"	Quit	Q "	noop

endmenu

newmenu gfe-cal

	menuline"	gfe cal menu	key "	noop
\u ss-cal	menuline"	Sun sensor cal	S "	ss-cal
\u lp-cal	menuline"	Langmuir probe cal	L "	lp-cal
\u pg-cal	menuline"	Pressure gauge cal	P "	pg-cal
\u tpm-cal	menuline"	Tpm calibration	T "	tpm-cal
\u cal-files	menuline"	cal File menu	f "	cal-files
	menuline"	Quit	Q "	noop

endmenu

newmenu exit-menu

```
menuline"    exit menu  key " menu-exit
menuline" esc      ESC " menu-exit
```

endmenu

newmenubar gsebar \ menubar for the pasp gse

```
\u ops-menu      +," Operations  "
\u dev-menu       +," Development "
\u cal-menu        +," Calibration "
\u setup-menu      +," Setup menu  "
\u exit-menu       +," esc=exit   "
```

endmenu

create gselist \ menu list for the pasp gse

```
\u ops-menu      ops-menu ,
\u dev-menu       dev-menu ,
\u cal-menu        cal-menu ,
\u setup-menu      setup-menu ,
\u exit-menu       exit-menu ,
```

: gmenu

```
( --- )
savecursor          \ save cursor position
['] mbutton save!> dobutton
cursor-off
nosetcur on
off> mrow
savescr              \ save original screen
save> mcol
on> mcol .menubar    \ display menubar without hilite
restore> mcol
```

%off> return-flg

begin

```
savescr              \ save it again
begin showmenus
key
    dup bl or 'q' = if return then
    dup 27 <>          \ while not esc
    over 13 <> and     \ and not carriage return
    ?menukey           \ or menu key
    if ?domkey
    then ?dup
until
```

```

showmenus

dup      13 =                \ is char a carriage return
if      drop
      mcol mrow dofunc \ then do the function
else    dup 27 =
      if      drop menu-exit \ discard if esc
      else    %on> ?browse
              doother \ else process the key
              %off> ?browse
      then
then      nosetcur off

restscr
return-flg until
restscr      \ recover original screen
restore> dobutton
restcursor

;

: gsecolumn  ( --- n1 )
      first.textcol 2+ ;

0 value gcolsave

: gsemenu  ( --- )
      gcolsave =: mcol      \ restore my column
      savemenu              \ save previous menu parms
      gsebar =: menubar    \ set pointer to menubar
      gselist =: menulist  \ and to the menulist
      ['] statusline is mline \ set the menu line func
      ['] gsecolumn is mcolumn \ and the menu column func
      ['] doachar is doother \ and the func to handle

keys

      ['] mbutton save!> sbutton \ not on the menu.
                                      \ install normal menu

handler

      %off> exit-flg
      begin
      gmenu
      exit-flg
      until
      restore> sbutton \ call the menu procedure
                          \ restore editor menu

handler

      restmenu \ restore previous menu

parms

      mcol =: gcolsave ; \ save my column for later

```

```
: vu2 2sec-cnt @ 0 20 at 5 u.hr many ;
```

```
: tt cls
  set-2sec
  up @ wake
  multi
  do-poll wake
  gsemenu
  do-poll sleep single
  clr-2sec ;
```

```
\ ' gsemenu is gseesc
```

headers

also forth definitions

```
: gse-start
  SPO @ 'TIB !
  >IN OFF
  SPAN OFF
  #TIB OFF
  LOADING OFF
  ONLY FORTH ALSO DEFINITIONS
  DEFAULTSTATE
  DEFAULT
  p_gse
  cls
  set-2sec
  up @ wake
  multi
  do-poll wake
\ gsemenu
\ do-poll sleep single
\ clr-2sec
;
```

```
\ ' gse-start is boot
```

```
  up @ sleep
  fsave gse.exe
\s
```

forth definitions

```
: (S [COMPILE] ( ; IMMEDIATE
```

```
: alloc-mem ( addr,n -- )
  over @ 0 =
  if dup u16/ swap $10 mod if 1+ then alloc 0 =
  if nip swap ! else true abort" mem alloc error" then
```

```

else 2drop then ;

: dealloc-mem ( addr -- )
  dup @ 0 = not
  if dup @ dealloc
    if true abort" mem alloc error" else off then
  else drop then ;

: u.hr ( n,#dig -- )
  base @ >r hex u.r r> base ! ;

```

APPENDIX L - DATA DICTIONARY

This data dictionary contains three main sections: data markers, event markers, and low level data definitions. The data markers and event markers are formatted using indents to show dependency of the levels of concatenation of low level words. Data obtained from testing or operating the PASP Plus instrument can be interpreted either manually or by a computer by use of this dictionary.

If a simple capture file is obtained from the transmitted data then there will be considerable data redundancy since the buffer will be padded with old data if enough new data is unavailable. This excess data can be stripped from the capture file data to make a buffer file. In either case there are always counters attached to the data so that only the necessary data is available for analysis.

The best way to use the data dictionary is to locate a data or event marker of interest and then look at the concatenated low level data words. Some data elements are included in a loop defined by double braces, {{ }}, indicating that the data will be taken the number of times shown by the numbers preceding the braces.

Each data marker and event marker has a number beside it which is the total number of bytes for that particular item.

DACS DATA DICTIONARY

I. Data Markers

1. Housekeeping Data Marker

: hk-v1 (-- 105)

"hsk"

hsk-cnt +

pclk +

hk-health-v1 +

hk-cal-v1 +

hk-status-v1 +

pg-v1 + ;

: hk-health-v1 (-- 41)

pm1

pm2 +

temp2-1 +

press2 +

temp1-1 +

press1 +

temp2-2 +

temp1-2 +

err-cnt +

16 0 {{ err-code + }} + ;

: hk-cal-v1 (-- 22)

ad1a-cal

ad1b-cal +

ad2a-cal +

ad2b-cal +

rtd6 +

rtd7 +

pclk +

met-clk + ;

: hk-status-v1 (-- 32)

res-cnt

wdg-cnt +

orb-cnt +

seg-cnt +

ele-cnt +

sby-cnt +

gar-cnt +

occ-cnt +

tss-cnt +

tst-cnt +

tim-cnt +

hsk-cnt +

ivs-cnt +

hvs-cnt +

DACS DATA DICTIONARY

```
tph-cnt +  
lmp-cnt +  
cal-cnt +  
cmd-cnt +  
int-cnt +  
tpm-cnt +  
met-cnt +  
oci-cnt +  
new-par +  
new-cal +  
new-cmd +  
s-ptr +  
r-ptr + ;
```

```
: pg-v1    ( -- 2)  
pg-press  
pg-temp + ;
```

2. IV Data Marker

```
: iv-v1    ( -- 111)  
  "iv" "#"  \note # = 1<= n <= 6 .  
ivs-cnt +  
pclk +  
ss-x +  
ss-y +  
ss-ata +  
array-temp +  
32 0 {{ iv-data + }} +  
array-temp + ;
```

```
: iv-data   ( -- 3)  
dig-cmd0  
array-v +  
array-i + ;
```

3.a. High Voltage Data Marker

```
: hv-v1     ( -- 1171)  
  "hvs"  
hvs-cnt +  
pclk +  
dig-cmd1 +  
dig-cmd2 +  
20 0 {{ hv-step + tpm-v1 + }} +  
emitter-i + ;
```

3.b. Short High Voltage Data Marker (short test sequence)

```
: hv-v2     ( -- 127)  
  "hvs"  
hvs-cnt +
```

DACS DATA DICTIONARY

```
pclk +
dig-cmd1 +
dig-cmd2 +
2 0 {{ hv-step + tpm-v1 + }} +
emitter-i + ;
```

```
    : hv-step    ( -- 4)
    etelem
    itelem +
    elect-i +
    ioc/pol + ;
```

Transient Pulse Monitor Data Marker

```
    : tpm-v1    ( -- 54)
```

```
    "tpm"
```

```
    tph-cnt +
```

```
    pclk +
```

```
    tpm-stat +
```

```
    tpm-data0 +
```

```
    tpm-stat +
```

```
    tpm-data1 + ;
```

```
    : tpm-data0    ( -- 22)
                  22 bytes; TPM format
    : tpm-data1    ( -- 22)
                  22 bytes; TPM format
```

4. Langmuir Probe Data Marker

```
    : lmp-v1    ( -- 451)
```

```
    "lmp"
```

```
    lmp-cnt +
```

```
    pclk +
```

```
    lp-temp +
```

```
    40 0 {{ lp-data + }} +
```

```
    lp-sweep +
```

```
    lp-temp + ;
```

```
    : lp-data    ( -- 11)
```

```
    lp-sweep
```

```
    10 0 {{ lp-ne + }} + ;
```

5. Calibration Marker

```
    : cal-v1    ( -- 210)
```

```
    "cal"
```

```
    cal-cnt +
```

```
    pclk +
```

```
    cal-id +
```

```
    cal-par +
```

```
    100 0 {{ cal-data + }} + ;
```

DACS DATA DICTIONARY

II. Event Markers

1. Reset Header

```
: reset-v1    ( -- 10)
  "res"
  res-cnt +
  pclk +
  rdo-stamp +
  cmd-stamp + ;
```

2. Watchdog Marker

```
: w-dog-v1    ( -- 8)
  "wdg"
  wdg-cnt +
  pclk + ;
```

3. Orbit mode Marker

```
: orb-v1      ( -- 200110)
  "or" "#"    \note # = 1<= n <= 6 .
  orb-cnt +
  pclk +
  3 0 {{ seq-v1 + }} +
  occ-v1 + ;
```

```
  : seg-v1     ( -- 57674)
    "se" "#"   \note # = 1<= n <= 3 .
    seg-cnt +
    pclk +
    4 0 {{ ele-v1 + }} +
    elh-v1 + ;
```

```
  : ele-v1     ( -- 14111)
    "el" "#"   \note # = 1<= n <= 4 .
    ele-cnt +
    pclk +
    elh-v1 +
    11 0 {{ hv-v1 + }} + ;
```

```
  : elh-v1     ( -- 1222)
    hk-v1 +
    6 0 {{ iv-v1 + }} +
    lmp-v1 + ;
```

```
  : occ-v1     ( -- 27080)
    "occ"
    occ-cnt +
    pclk +
    48 0 {{ hk-v1 + lmp-v1 + 2 0 {{ hv-step + }} + }} + ;
```

DACS DATA DICTIONARY

4. Standby Mode Marker - for 10 cycles

: stby-v1 (-- 8+10(105) 1058)

"sby"

sby-cnt +

pclk +

10 0 {{ hk-v1 + }} + ;

5. Garrett Mode Marker - for 10 cycles

: gar-v1 (-- 8+10(564) 5648)

"gar"

gar-cnt +

pclk +

10 0 {{ hk-v1 + lmp-v1 + 0 2 0 {{ hv-step + }} + }} + ;

6. Short Test Sequence Marker

: tss-v1 (-- 1401)

"tss"

tss-cnt +

pclk +

elh-v1 +

9 0 {{ hv-v2 + }} + ;

7. Test Sequence Marker

: tst-v1 (-- 11769)

"tst"

tst-cnt +

pclk +

elh-v1 +

9 0 {{ hv-v1 + }} + ;

DACS DATA DICTIONARY

III. Low Level Data Definitions

\ byte = 8 bits

: res-cnt	(-- 1)	" res-cnt	" 1-byte.	;
: wdg-cnt	(-- 1)	" wdg-cnt	" 1-byte.	;
: orb-cnt	(-- 1)	" orb-cnt	" 1-byte.	;
: seg-cnt	(-- 1)	" seg-cnt	" 1-byte.	;
: ele-cnt	(-- 1)	" ele-cnt	" 1-byte.	;
: sby-cnt	(-- 1)	" sby-cnt	" 1-byte.	;
: gar-cnt	(-- 1)	" gar-cnt	" 1-byte.	;
: occ-cnt	(-- 1)	" occ-cnt	" 1-byte.	;
: tss-cnt	(-- 1)	" tss-cnt	" 1-byte.	;
: tst-cnt	(-- 1)	" tst-cnt	" 1-byte.	;
: tim-cnt	(-- 1)	" tim-cnt	" 1-byte.	;
: hsk-cnt	(-- 1)	" hsk-cnt	" 1-byte.	;
: ivs-cnt	(-- 1)	" ivs-cnt	" 1-byte.	;
: hvs-cnt	(-- 1)	" hvs-cnt	" 1-byte.	;
: tph-cnt	(-- 1)	" tph-cnt	" 1-byte.	;
: lmp-cnt	(-- 1)	" lmp-cnt	" 1-byte.	;
: cal-cnt	(-- 1)	" cal-cnt	" 1-byte.	;
: cmd-cnt	(-- 1)	" cmd-cnt	" 1-byte.	;
: int-cnt	(-- 1)	" int-cnt	" 1-byte.	;
: tpm-cnt	(-- 1)	" tpm-cnt	" 1-byte.	;
: met-cnt	(-- 1)	" met-cnt	" 1-byte.	;
: oci-cnt	(-- 1)	" oci-cnt	" 1-byte.	;
: cal-id	(-- 1)	" cal-id	" 1-byte.	;
: cal-par	(-- 1)	" cal-par	" 1-byte.	;
: pm1	(-- 1)	" pm1	" 1-byte.	;
: pm2	(-- 1)	" pm2	" 1-byte.	;
: temp2-1	(-- 1)	" temp2-1	" 1-byte.	;
: press1	(-- 1)	" press1	" 1-byte.	;
: temp1-1	(-- 1)	" temp1-1	" 1-byte.	;
: press2	(-- 1)	" press2	" 1-byte.	;
: temp2-2	(-- 1)	" temp2-2	" 1-byte.	;
: temp1-2	(-- 1)	" temp1-2	" 1-byte.	;
: err-cnt	(-- 1)	" err-cnt	" 1-byte.	;
: dig-cmd0	(-- 1)	" dig-cmd0	" 1-byte.	;
: dig-cmd1	(-- 1)	" dig-cmd1	" 1-byte.	;
: dig-cmd2	(-- 1)	" dig-cmd2	" 1-byte.	;
: ss-x	(-- 1)	" ss-x	" 1-byte.	;
: ss-y	(-- 1)	" ss-y	" 1-byte.	;
: ss-ata	(-- 1)	" ss-ata	" 1-byte.	;
: array-i	(-- 1)	" array-i	" 1-byte.	;
: array-v	(-- 1)	" array-v	" 1-byte.	;
: pg-press	(-- 1)	" pg-press	" 1-byte.	;
: pg-temp	(-- 1)	" pg-temp	" 1-byte.	;
: lp-temp	(-- 1)	" lp-temp	" 1-byte.	;

DACS DATA DICTIONARY

```
: lp-sweep      ( -- 1 ) " lp-sweep  " 1-byte. ;
: lp-ne         ( -- 1 ) " lp-ne     " 1-byte. ;
: tpm-stat      ( -- 1 ) " tpm-stat  " 1-byte. ;
: tpm-data      ( -- 1 ) " tpm-data  " 1-byte. ;
: emitter-i     ( -- 1 ) " emitter-i " 1-byte. ;
: etelem        ( -- 1 ) " etelem    " 1-byte. ;
: itelem        ( -- 1 ) " itelem    " 1-byte. ;
: elect-i       ( -- 1 ) " elect-i   " 1-byte. ;
: ioc/pol       ( -- 1 ) " ioc/pol   " 1-byte. ;
: rdo-stamp     ( -- 1 ) " rdo-stamp " 1-byte. ;
: cmd-stamp     ( -- 1 ) " cmd-stamp " 1-byte. ;
```

\ word = 2 bytes = 16 bits

```
: new-par       ( -- 2 ) " new-par   " 2-byte. ;
: new-cal       ( -- 2 ) " new-cal   " 2-byte. ;
: new-cmd       ( -- 2 ) " new-cmd   " 2-byte. ;
: s-ptr         ( -- 2 ) " s-ptr     " 2-byte. ;
: r-ptr         ( -- 2 ) " r-ptr     " 2-byte. ;
: cal-data      ( -- 2 ) " cal-data  " 2-byte. ;
: err-code      ( -- 2 ) " err-code  " 2-byte. ;
: ad1a-cal      ( -- 2 ) " ad1a-cal  " 2-byte. ;
: ad1b-cal      ( -- 2 ) " ad1b-cal  " 2-byte. ;
: ad2a-cal      ( -- 2 ) " ad2a-cal  " 2-byte. ;
: ad2b-cal      ( -- 2 ) " ad2b-cal  " 2-byte. ;
: rtd6          ( -- 2 ) " rtd6      " 2-byte. ;
: rtd7          ( -- 2 ) " rtd7      " 2-byte. ;
: array-temp    ( -- 2 ) " array-temp" 2-byte. ;
```

\ double word = 4 bytes = 32 bits

```
: pclk          ( -- 4 ) " pclk      " 4-byte. ;
```

\ triple word = 6 bytes = 48 bits

```
: met-clk       ( -- 6 ) " met-clk   " 6-byte. ;
```